

Palvelun laatua parantava integraatioarkkitehtuuri

Markus Salomaa

Tampereen yliopisto
Informaatiotieteiden yksikkö
Tietojenkäsittelyoppi
Pro gradu -tutkielma
Ohjaaja: Mikko Ruohonen
Joulukuu 2014

Tampereen yliopisto

Informaatiotieteiden yksikkö

Tietojenkäsittelyoppi

Markus Salomaa: Palvelun laatua parantava integraatioarkkitehtuuri

Pro gradu -tutkielma, 59 sivua

Joulukuu 2014

Tässä tapaustutkimuksessa seurattiin järjestelmäintegraatioprojektia, jossa uudistettiin teollisuuden kunnossapitopalveluita tarjoavan yrityksen integraatioarkkitehtuuri. Seurattavan projektin lopputuloksena syntyi palvelukeskeisen arkkitehtuurin periaatteiden mukaisesti toteutettu ratkaisu, jonka ytimen muodosti palveluväylä. Palveluväylän yhteyteen rakennettiin perustietojärjestelmä organisaation toiminnalle tärkeiden tietojen hallintaan.

Tutkielman tavoitteena oli selvittää millainen nykyaikaisen integraatioarkkitehtuurin tulisi olla, jotta sen avulla voitaisiin saavuttaa palvelun laadun paranemista organisaatiossa, joka solmii lukumäärältään paljon yritykseltä-yritykselle -liiketoimintasuhteita. Arkkitehtuurin ja toteutettujen ratkaisujen piti olla sellaisia, minkä avulla pystyttäisiin tukemaan organisaation kasvua ja laajentumista. Tapaustutkimus ja lopputuloksena syntynyt integraatioarkkitehtuuri kuvattiin suunnittelutieteellisen tutkimusotteen periaatteiden mukaisesti.

Avainsanat ja -sanonnat: järjestelmäintegraatio, integraatioarkkitehtuuri, palvelukeskeinen arkkitehtuuri, palveluväylä, B2B-integraatio.

Sisällys

1. Johdanto	1
2. Tutkimus	4
3. Palvelun laatu.....	7
3.1. Tietojärjestelmien onnistuminen laadun mittarina	8
4. Järjestelmäintegraatio yleisesti	11
4.1. Integraation hyödyt	12
4.2. Integraation vaatimukset	13
5. Yleisimmät integraatiotavat ja -mallit	15
5.1. Point-to-point.....	15
5.2. Hub-and-spoke	17
5.3. Message Bus.....	19
6. Palvelukeskeinen arkkitehtuuri.....	21
6.1. Arkkitehtuurin pääpiirteet	22
6.2. Palvelu	23
6.3. Yhteentoimivuus	25
6.4. Löyhät sidokset	25
6.5. Palveluväylä.....	26
6.5.1. Palveluväylän tehtävät.....	26
6.5.2. Palveluväylän tekniset toteutustavat	27
6.5.3. Palveluväylän hyödyt.....	28
6.6. Kanoninen malli	29
6.7. BAM	30
7. Liiketoimintaprosessit.....	32
7.1. Liiketoimintaprosessien viestintästandardit.....	32
7.2. Integraatioarkkitehtuuri osana liiketoimintaprosesseja	33
8. Tapaustutkimus - teollisuuden kunnossapitopalveluita tarjoava yritys.....	36
8.1. Asiakkaan lähtötila	36
8.1.1. Edeltävän järjestelmäarkkitehtuurin vaikutukset liiketoimintaan	37
8.1.2. Tietojärjestelmiin liittyneet ongelmat.....	39
8.2. Asiakkaan tavoitteet projektille	40
8.3. Haasteet integraatioarkkitehtuuria muutettaessa	42
8.4. Toteutunut arkkitehtuuri	43
8.4.1. Perustieto.....	44
8.4.1.1 Perustieto osana integraatioarkkitehtuuria	45
8.4.2. Palveluväylä integraatioarkkitehtuurin keskiössä.....	46

8.5. Mitä integraatioarkkitehtuurilla saatiin aikaan?	48
8.6. Ennakoimattomat vaikutukset ja mahdolliset parannusehdotukset...	53
9. Yhteenveto	55
Viiteluettelo	57

1. Johdanto

Käytössä olevien tietoteknisten sovellusten ja tietojärjestelmien lukumäärä on kasvanut räjähdysmäisesti 15:n viime vuoden aikana. Suuria, kaikenkattavia tietojärjestelmiä tehdään aikaisempaa selvästi vähemmän, koska tilalle on tullut suuntaus, jossa yksittäiset sovellukset saavat vastuulleen ainoastaan yhden, selkeämmin rajatun osakokonaisuuden. Sovellukset taas koostuvat erillisistä, tiivistä toistensa kanssa kommunikoivista komponenteista [Tähtinen, 2005]. Tämä sovellusten sisäisen ja ulkoisen kommunikaation kasvu on puolestaan synnyttänyt tarpeen järjestelmäintegraatiolle, jonka avulla hajautetun sovellusarkkitehtuurin osakokonaisuudet saadaan toimimaan yhdessä toivotulla tavalla.

Tämän päivän mediassa puhutaan paljon järjestelmäintegraatiosta siten kuin se olisi yksittäinen teknologinen tuote tai sovellus. Tätä integraatio ei kuitenkaan missään mielessä ole, vaan integraatio pitäisi nähdä kokoelmana erilaisia ajatus- ja suunnittelumalleja [Tähtinen, 2005]. Osa näistä malleista on jo tietotekniikan historiaankin peilaten hyvin vanhoja, mutta silti niitä käytetään osana nykypäivän ohjelmistokehitystä. Toisaalta hyvin monessa nykyaikaisessa organisaatiossa tietoteknisiä laitteita ja järjestelmiä on jo niin paljon, että vanhat ja yksinkertaiset mallit eivät ole enää tarpeeksi kattavia alati muuttuviin tarpeisiin. Siksi on syntynyt tarve kehittää erilaisia integraatioarkkitehtuureja ja -malleja, joiden avulla yritykset pystyvät hyödyntämään käytössä olevaa tietotekniikkaansa mahdollisimman laajasti. Siten yrityksillä on mahdollisuus saavuttaa kilpailuetua ja palvella asiakkaitaan paremmin.

Millainen nykyaikaisen integraatioarkkitehtuurin sitten pitäisi olla, jotta sillä voidaan hallita nopeasti muuttuvia liiketoimintatarpeita, ja tukea alati kehittyviä teknologisia ratkaisuja? Tässä työssä olen yrittänyt saada vastauksen tähän kysymykseen, sillä työn teon aikana olin mukana integraatioprojektissa, jossa teollisuuden palveluyritys päätti

nykyaikaistaa valtaosan tietojärjestelmistään osana uudistunutta strategiaansa. Tarkasteltavan yrityksen liiketoiminnalle on ominaista, että organisaatio solmii lukumäärältään paljon yritykseltä-yritykselle -liiketoimintasuhteita, mikä taas omalta osaltaan asettaa vaatimuksia myös teknologisille ratkaisuille. Projektin keskeisimpänä kokonaisuutena oli toteuttaa integraatioarkkitehtuuri, jonka avulla pystyttäisiin tukemaan yrityksen uudistunessa strategiassa määriteltyjä kasvu- ja laajentumistavoitteita.

Tässä tutkielmassa tuon esille yhden vaihtoehdon integraatioarkkitehtuurista, joka toteutettiin tarkasteltavan projektin aikana. Kirjallisuuskatsauksen perusteella erilaisia integraatiomalleja tai toimintatapoja on käsitelty akateemisessa kirjallisuudessa, mutta varsinaisia integraatioarkkitehtuureja koskevia tekstejä löytyy lähinnä yritysmaailman puolelta. Arkkitehtuurin lisäksi tarkoituksena on myös tuoda esille niitä motiiveja, joiden kautta voidaan perustella investointeja integraatiokerroksen parantamiseksi. Tämän lisäksi tutkielmassa haluttiin selvittää integraatioarkkitehtuurin merkitystä liiketoiminnan kasvun ja kilpailukyvyn tukijana. Koko tutkimusta eteenpäin ajavana tutkimuskysymyksenä on ollut: Millainen integraatioarkkitehtuuri mahdollistaisi yritysten välisessä liiketoiminnassa palveluiden laadun paranemisen?

Tutkielma koostuu kahdesta osasta, kirjallisuuskatsauksesta ja tapaustutkimuksesta. Kirjallisuuskatsauksessa luodaan käsitys siitä, mitä järjestelmäintegraatio on ja millä eri tavoilla integraatioita voidaan toteuttaa. Lähdemateriaalia on saatu työnantajani hankkimasta alan kirjallisuudesta, Tampereen yliopiston kirjastosta ja sähköisistä kokoelmista ja Google Scholarista. Tapaustutkimuksen aluksi kuvataan kohteena olevan yrityksen lähtötila, aikaisemman tietojärjestelmäarkkitehtuurin ongelmat ja asiakkaan tavoitteet projektilta. Työn loppuosassa peilataan näitä tavoitteita syntyneeseen lopputulokseen ja tarkastellaan päästiinkö asetettuihin tavoitteisiin.

Tutkielman kirjallisuuskatsaus sisältyy lukuihin 3-7. Kolmannessa luvussa käydään läpi palvelun laadun käsite, ja se miten tätä pitäisi tarkastella tietojärjestelmien ja

integraatiototeutusten osalta. Neljännessä luvussa käydään läpi lyhyesti järjestelmäintegraation historia, erilaiset integraatiotyypit ja se, millaisista lähtökohdista integraatiota yleensä aloitetaan toteuttaa. Tässä luvussa nostetaan esille myös ne hyödyt, joita organisaatio voi onnistuneella järjestelmäintegraatiolla saavuttaa. Neljännessä luvussa käydään myös karkealla tasolla läpi vaatimukset, joita järjestelmäintegraation onnistuminen edellyttää. Työn viides luku esittelee järjestelmäintegraatioissa yleisimmin käytetyt integraatiomallit. Samalla tarkastellaan eri mallien vahvuudet, mutta myös ongelmakohtia.

Kuudes luku tarkastelee palvelukeskeisen arkkitehtuurin käsitettä. Luvussa käydään läpi arkkitehtuurin peruspiirteet, yleisimmät käsitteet, kanoninen tietomalli, BAM-ratkaisu (Business activity monitoring), ja erityisesti palveluväylän käsite. Lisäksi syvennyttään palveluväylän tehtäviin integraatioarkkitehtuurin operoijana, yleisiin toteutusperiaatteisiin, teknisiin ratkaisuvaihtoehtoihin ja mahdollisiin hyötyihin, joita palveluväylällä voidaan saavuttaa. Työn seitsemännessä luvussa avataan liiketoimintaprosessin käsite ja se, miten prosesseja voidaan kuvata nykyaikaisten standardien avulla.

2. Tutkimus

Tutkimukseni tutkimusotteeksi valikoitui tapaustutkimus. Tapaustutkimus on enemmänkin tutkimusstrategia, sillä sen avulla pystytään luonnollisesti vastaamaan siihen mitä tutkitaan [Cassel and Symon, 2004]. Kuten aikaisemmin on todettu, sain olla tutkimuksen teon aikana mukana suurehkossa integraatiohankkeessa, jossa uudistettiin teollisuuden palveluita tarjoavan yrityksen integraatioarkkitehtuuri. Integraatioarkkitehtuureista ei kuitenkaan löydy kovin paljon akateemista tutkimusta, joten tapaustutkimus oli sopiva valinta. Tapaustutkimus soveltuu hyvin tutkimusaiheeseen, josta on tehty vain vähän empiiristä tutkimusta [Cassel and Symon, 2004].

Kaiken kaikkiaan tuntui luontevalta kuvata ja reflektoida integraatiohankkeen aikana tapahtuneita asioita ja pohtia niiden syy-seuraussuhteita. Tämän tyyppistä tutkimusmetodia kutsutaan myös refleктоivaksi käytännöksi [Schön, 1983]. Metodin keskeisenä ajatuksena on, että uuden oppimista voi tapahtua käytännön elämässä havainnoinnin kautta, peilaamalla jo koettuja kokemuksia tuleviin tapahtumiin [Schön, 1983]. Tällaisissa tilanteissa oma osaaminen kehittyy ja ongelmia voi ratkaista muun muassa improvisoinnin kautta.

Yin [1989] puolestaan toteaa, että tapaustutkimus on empiirinen tutkimustapa, jossa tutkitaan reaaliaikaisia ilmiöitä niiden todellisessa kontekstissa. Tästä johtuen tapaustutkimukselle on tyypillistä, että ilmiön ja kontekstin rajapinta ei ole selvä ja käytettävissä olevia tiedon lähteitä on paljon. Tapaustutkimuksessa siis tutkitaan nykyajan ilmiötä sen todellisessa käyttöympäristössä. Yin [1989] myös erottelee tapaustutkimuksen muista tutkimusotteista tutkimuksen kysymyksen muodon, tutkittavien käyttäytymisen kontrollin ja ajanhetken perusteella. Tapaustutkimuksen

tueksi toteutettiin myös kirjallisuuskartoitus, jossa käytiin läpi alan kirjallisuudessa vakiintuneita integraatiokäytäntöjä ja toteutusmalleja. Tämän kartoituksen tuloksia peilataan tapaustutkimuksen lopputulosten kanssa.

Tapaustutkimuksen laadun varmistamiseksi on keskeistä kerätä laadukasta tietoa riittävän kattavasti [Yin, 1989]. Tässä tutkielmassa tietoja ja näytöjä on kerätty projektin aikana tehdyistä dokumenteista, vapaalla ja osallistuvalla havainnoinnilla sekä haastatteleamalla muita projektiin osallistuneita henkilöitä. Päädyin yhden tapauksen tutkielmaan, koska aihepiiri, projektin laajuus ja vaativuus eivät mahdollistaneet useamman tapauksen havainnointia. Valitun tutkimusmenetelmän haasteena on ollut kontrolloitujen havaintojen tekeminen. Kuten tapaustutkimuksen luonteeseen kuuluu, tutkimustyyppi tuottaa paljon muuttujia, mutta verrattain vähän havaintoarvoja, joten tilastollisia tai kvantitatiivisia päätelmiä on ollut vaikea tehdä [Järvinen ja Järvinen, 1996].

Tässä tutkimuksessa varsinaista tapaustutkimuksen osuutta käydään läpi suunnittelutieteellisen tutkimusotteen perusperiaatteiden mukaisesti. Suunnittelutieteellinen tutkimusote on Hevnerin ja muiden [2004] kehittämä ongelmanratkaisuparadigma, joka karrikoiduna pyrkii tuottamaan hyödyllisiä toteutuksia, artefakteja, oikeisiin liiketoimintaongelmiin. Keskeistä ajattelutavalle on, että sen avulla toteutettujen ratkaisuiden hyödyllisyys arvioidaan erilaisilla artefaktin ominaisuuksia arvioivilla mittareilla: onko se toiminut, täyttänyt tavoitteensa ja tuottanut oikeita liiketoimintahyötyjä [Hevner *et al.*, 2004]. Arviointi voi kuitenkin olla haasteellista, sillä siihen vaikuttaa merkittävästi ympäristö, jossa toteutettu järjestelmä toimii. Suunnittelutieteellistä tutkimusotetta voidaanakin kutsua ”rakenna ja arvio” -paradigmaksi.

Hevner ja muut [2004] määrittelevät tutkimuksessaan seitsemän vaihetta sisältävän ohjeen tietojärjestelmätieteen suunnittelutieteelliselle tutkimukselle.

1. Artefaktin suunnittelu. Tutkimuksen täytyy tuottaa johonkin organisaation merkittävään ongelmaan rakennettu tietojärjestelmäartefakti.
2. Painota ajankohtaisuutta ja relevanssia. Tutkimuksen tavoitteena on hankkia tietämystä ja ratkaisuja teknologiapainotteisiin liiketoimintaongelmiin.
3. Arvioi. Toteutetun artefaktin hyöty ja laatu pitää osoittaa arvioinnin kautta. Arviointiperusteet muodostuvat kulloisenkin tilanteen liiketoimintaympäristöstä ja sen kautta tulevista vaatimuksista.
4. Tutkimuksen kontribuutiot. Toimiva suunnittelutieteellinen tutkimus tuottaa uutta tietoa toteutettujen artefaktien, suunnittelumetodien ja suunnitteluperusteiden kautta.
5. Tieteellinen tarkkuus ja perusteellisuus. Onnistunut lopputulos voidaan saavuttaa soveltamalla tarkasti määriteltyjä metodeja sekä rakennus- että arviointivaiheessa.
6. Tutkimusratkaisuiden etsintäprosessi. Onnistuneen artefaktin suunnittelu on etsintäprosessi, jonka lopputuloksena saavutetaan tehokas ratkaisu kyseessä olevaan ongelmaan. Suunnitteluvaihe on luonteeltaan iteratiivista, joten ratkaisun laatu paranee suunnittelun edetessä.
7. Tulosten esittely ja kommunikointi. Suunnittelutieteelliset tutkimustulokset pitää esitellä sekä teknilliselle että myös johtamissuuntautuneelle yleisölle.

Tässä tutkimuksessa esitellään kappaleessa 8 tapaustutkimus, jossa on pyritty noudattamaan edellä kuvattuja ohjeistuksia.

3. Palvelun laatu

Palvelun laatu on käsite, joka on synnyttänyt akateemista keskustelua jo vuosikymmenten ajan, sillä sitä on jo luonteensa puolesta hyvin vaikea määritellä tai mitata [Wisniewski, 2001]. Erilaisia määritelmiä on kuitenkin yritetty tehdä, ja pääsääntöisesti näissä määritelmissä palvelun laadulla on tarkoitettu henkilökohtaisen ennakko-odotuksen tai tarpeen ja reaali maailmassa koetun kokemuksen välistä suhdetta [Shanin and Samea, 2010]. Ennakko-odotukset voivat koostua muun muassa aiemmista kokemuksista, brändin maineesta, suositteluista ja hinnasta. Kokemukset taas muodostuvat esimerkiksi ympäristön ja palvelun teknisen ja toiminnallisen laadun perusteella. Voidaankin todeta, että kun ennakko-odotukset ovat korkeammat kuin varsinainen toteutunut palvelu, niin koettu laatu on vähemmän tyydyttävää. Tämä voi taas ilmentyä asiakastyytymättömyytenä.

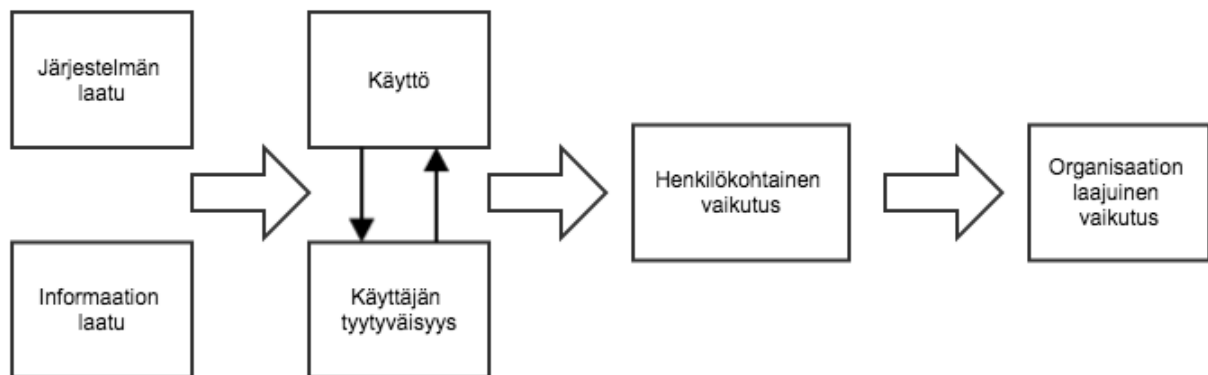
Palvelun laadun mittaamista on pyritty helpottamaan kehittämällä erilaisia mittaustyökaluja. Yksi tällaisista on alalla jo hyvin vakiintuneen aseman saanut SERVQUAL-malli. Työkalu on kehitetty jo 1980-luvun puolivälissä ja sen tavoitteena on tarjota vaihtoehtoja palvelun laadun mittaamiseen ja hallintaan. SERVQUAL-mallissa palvelun laadun mittaaminen perustuu viiteen erilaiseen tekijään: konkreettisiin tekijöihin, luotettavuuteen, reagoimiskykyyn, varmuuteen ja empatiaan. [Wisniewski, 2001].

Perinteiset palvelun laadun määritelmät ja mittarit ovat kuitenkin tulleet ongelmallisiksi tai osittain puutteelliseksi, kun tarkastelun kohteina ovat olleet tietojärjestelmät ja niiden tarjoamat palvelut. Perinteisten mallien toimimattomuus ei kuitenkaan ole mikään yllätys, sillä ne ovat aikanaan syntyneet tarpeeseen, jossa on haluttu mitata esimerkiksi ruokakaupassa koetun palvelun laatua. Jotta osattaisiin valita oikeanlaiset työkalut palvelun laadun mittaamiseen, on tärkeä määritellä, mitä kulloisessakin tilanteessa

palvelulla tarkoitetaan. On eri asia puhua esimerkiksi verkkokaupan tarjoamista palvelusta, kun palvelutermillä voidaan yhtälailla tarkoittaa toteutusta, jolla tietojärjestelmät välittävät tietoa automatisoidusti toisilleen. Tällaisessa tilanteessa varsinaista loppukäyttäjän kokemusta ei siis ole lainakaan mukana, joten laadun mittaaminen vaikeutuu entisestään.

3.1. Tietojärjestelmien onnistuminen laadun mittarina

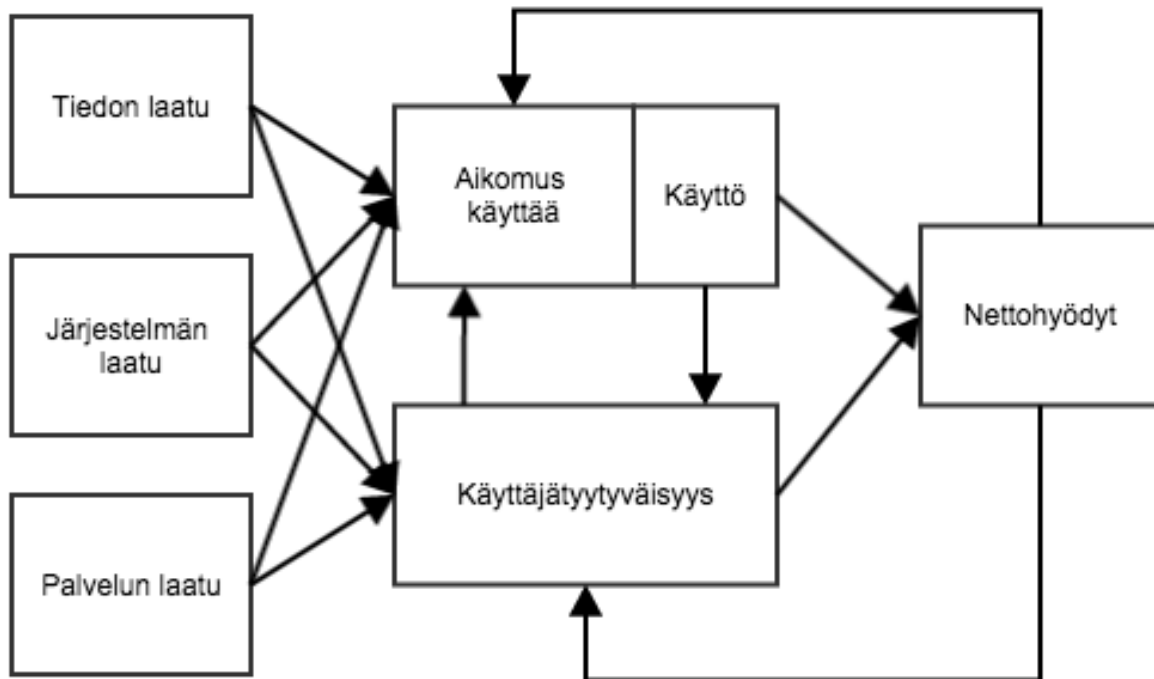
Osittain aikaisemmin kuvattujen haasteiden vuoksi, tietojärjestelmien, ja erityisesti operatiivisia järjestelmiä tukevien järjestelmien, laatua ja onnistumista varten on pitänyt kehittää uudentyyppisiä malleja ja mittareita. Yksi tällaisista on perinteinen DeLone ja McLeanin [1992] tietojärjestelmien onnistumismalli (Kuva 1.).



Kuva 1. DeLonen ja McLeanin [1992] tietojärjestelmien onnistumismalli

Mallin ajatuksena on, että järjestelmäprojektin onnistuminen riippuu toteutetun järjestelmän ja käytettävän informaation laadusta. Niiden laatu heijastuu sellaisenaan järjestelmän käyttöön ja käyttäjätyytyväisyyteen. Käyttäjätyytyväisyys on käyttäjänsä henkilökohtainen kokemus, mutta yhtenevät mielipiteet heijastuvat organisaatiotasolle. DeLonen ja McLeanin mukaan käyttäjätyytyväisyyden merkityksen suhde kasvaa muihin mittareihin verrattuna silloin, kun järjestelmä on valmis ja käyttäjät pääsevät käyttämään

sitä. Lopulliseen tietojärjestelmää kohtaan koettuun tyytyväisyyteen vaikuttavat myös muut organisaation asiat, muun muassa johdon tyytyväisyys [DeLone and McLean, 2003].



Kuva 2. DeLonen ja McLeanin [2003] tietojärjestelmien onnistumismalli

Alkuperäisestä, jo klassikoksi muodostuneesta mallista on kehitetty myös päivitetty versio (Kuva 2.), jossa palvelun laatu on nostettu yhdeksi keskeiseksi tietojärjestelmien onnistumistekijäksi. Päivitetyn mallin perusajatus on kuitenkin pääpiirteittäin sama kuin mitä alkuperäisessä mallissa. Mallin mukaan tietojärjestelmästä voidaan arvioida tiedon laatua, järjestelmän laatua ja palvelun laatua. Nämä kolme osa-aluetta vaikuttavat siihen, miten järjestelmää käytetään ja kuinka tyytyväisiä sen käyttöön ollaan. Positiivisesta käyttäjätyytyväisyydestä ja tietojärjestelmien aktiivisemmasta käytöstä syntyy nettohyötyjä, jotka puolestaan vaikuttavat käyttäjätyytyväisyyteen ja tuleviin käyttökertoihin [DeLone and McLean, 2003].

DeLone ja McLeanin tietojärjestelmien onnistumismallit on tarkoitettu sellaisten tietojärjestelmien onnistumisen ja laadun arviointiin, joissa vuorovaikutus tapahtuu

suoraan loppukäyttäjän kanssa. Tämä ilmenee erityisesti palvelun laatu- ja käyttäjätyytyväisyyskäsitteiden kautta, joilla voidaan esimerkiksi kuvata miten helppokäyttöinen verkkokauppa on. Siksi integraatioprojekteilla saavutettuja vaikutuksia onkin hyvä tarkastella koko organisaation tietojärjestelmäkentän tasolla, sillä integraatioarkkitehtuureihin liittyvät ratkaisut ja toteutukset ovat lähtökohtaisesti vain operatiivisia järjestelmiä tukevia kokonaisuuksia. Varsinaiset hyödyt ilmenevät siten, että liiketoimintaprosessit, ja niitä tukevat operatiiviset järjestelmät, toimivat aikaisempaa tehokkaammin.

4. Järjestelmäintegraatio yleisesti

Järjestelmäintegraatiolla tarkoitetaan toimintaa, jossa organisaation sisälle hajautunut tieto yhdistetään, ja näin aikaisemmin keskenään yhtyeensopimattomat sovellukset saadaan kommunikoimaan toistensa kanssa. [Tähtinen, 2005]. Teollisuudessa tämä voi tarkoittaa esimerkiksi toiminnanohjaus- ja varastonhallintajärjestelmän yhdistämistä siten, että tieto liikkuu järjestelmien välillä. Yhteisen tiedon ollessa sähköisesti saatavilla, järjestelmistä syntyy hallittavampi kokonaisuus ja tuotantoa on helpompi ohjata. Näin ollen järjestelmäintegraatiosta puhuttaessa ei siis tarkoiteta yksittäistä tuotetta, vaan ajattelutapaa, jonka avulla pystytään luomaan yritykselle mahdollisimman tehokas liiketoimintaympäristö.

Järjestelmäintegraation historia juontaa juurensa 1950-luvulle, jolloin käynnistettiin projektit SAGE (Semi-Automatic Ground Environment) ja SABRE (Semi-Automatic Business Research Environment). SAGE oli Yhdysvaltojen ilmavoimien projekti, jonka tarkoituksena oli kehittää automatisoitu ilmapuolustusjärjestelmä kylmän sodan tarpeisiin. SABRE taas oli American Airlines -lentoyhtiön kehittämä järjestelmä, jonka avulla hoidettiin lentojen paikanvaraus ja lipunmyynti. Kumpaakin projektia jatkokehitettiin, ja ne synnyttivät lukuisia vastaavia projekteja eri yrityssektoreille. SABRE -järjestelmä on yhä lentoyhtiöiden käytössä, ja se on yksi ilmailualan suurimmista tietojärjestelmistä [Tähtinen, 2005].

Karkeasti jaoteltuna järjestelmäintegraatiot voidaan jakaa sisäisiin ja ulkoisiin integraatioihin. Sisäiset integraatiot, Enterprise Application Integration (EAI), tarkoittavat tiedon siirtoa yrityksen sisällä olevien järjestelmien kesken. Yritysten ulkopuolisilla integraatioilla, business-to-business integration (B2B), taas tarkoitetaan integraatioita, joita toteutetaan yritysten väliseen tiedonsiirtoon. B2B- integraatioissa järjestelmien pitää

pystyä kommunikoimaan keskenään, jotta tiedonsiirto toimisi ja olisi saumatonta. Tämän toteutukseen voidaan tehdä räätälöityjä ratkaisuja, mutta nykyään yleiset sanomanvälitysstandardit ovat yleistyneet ja tehneet tiedonsiirrosta helpompaa.

Tietojärjestelmien integrointi on aina kohdealueelleen tärkeä projekti, ja isojen projektien toteutus voi viedä useita vuosia. Tämän vuoksi integraation suunnittelu pitää tehdä huolella ja kauaskantoisesti. Lopputuloksen kannalta on tärkeää tehdä heti alusta selväksi, miltä kannalta integraatiota lähestytään. Lähestyminen voi tapahtua esimerkiksi organisatorisesta tai teknillisestä näkökulmasta. Integraatio voi tapahtua myös joko horisontaalisesti tietyn hierarkiatason sisällä tai vertikaalisesti eri hierarkiatasojen välillä. Perinteisesti integraatio on jaettu vielä useampiin kerroksiin, kuten data-, viesti- ja prosessitasoon, jotta kokonaisuudesta tulisi helpommin hahmotettavampi [Mykkänen *et al.*, 2003].

4.1. Integraation hyödyt

Tietojärjestelmän ja sen integroinnin hyödyllisyyttä voidaan kuvata ns. Metcalfen lailla, jonka mukaan verkon arvo on verrannollinen verkon käyttäjien lukumäärän neliöön [Tähtinen, 2005]. Lain perusteella voidaankin päätellä, että tietojärjestelmäverkko, jossa on kuusi erillistä käyttäjää, on yhdeksän kertaa arvokkaampi kuin verkko, jossa on vain kaksi käyttäjää.

Kahden järjestelmän välille tehty erillISRatkaisu on useasti vain väliaikaisratkaisu ja aiheuttaa turhaan eri liittymien määrän kasvua. Tämän vuoksi pitkäjänteinen integraatioarkkitehtuurin suunnittelu on toivottavaa heti alusta lähtien. Kasvava verkko luo myös kiinnostusta verkon ulkopuolella oleville palveluille, mistä syystä verkko voi kasvaa suunnittelua nopeammin. Joustava tietojärjestelmä tarjoaa mahdollisuuden liiketoimintaprosessin tai organisaation muutoksiin sekä vähentää riippuvuutta ohjelmistotoimittajista [Tähtinen, 2005]. Järjestelmien integraatiolla pystytään myös

jalostamaan kerran syötettyä tietoa, mikä auttaa hallitsemaan prosessia paremmin. Tämä tukee päätöksentekoa ja antaa yrityksen johdolle mahdollisuuden saada kokonaisnäkemys yrityksen tilasta, muun muassa raportointityökaluilla. Hyvin suunniteltu tietojärjestelmäintegraatio selkeyttää myös tietojärjestelmäarkkitehtuuria ja näin muun muassa vähentää ylläpitokuluja.

Vuosien saatossa onkin tullut ilmi, että ilman järjestelmäintegraatiota ei ole ollut mahdollista muodostaa laajempia tietojärjestelmäkokonaisuuksia. Tähän ei ole tullut muutosta, vaikka www-tekniikat ovat kehittyneet viime vuosina isoin harppauksin. Tiedon jakamisen tarve on yhä kasvanut, ja yritysten järjestelmien välisestä tiedonsiirrosta on tullut entistä merkittävämpi osa liiketoimintaprosesseja.

4.2. Integraation vaatimukset

Kuten aikaisemmin on todettu, tietojärjestelmien integrointi ei tapahdu sormia napsauttamalla, vaan se voi olla hyvin pitkä ja vaikea prosessi erityisesti, jos integraation kohteena on esimerkiksi vaikeaselkoinen perinnejärjestelmä. Perinnejärjestelmällä tarkoitetaan tässä yhteydessä vanhentunutta tietojärjestelmää, joka on vielä käytössä, koska se on käyttöympäristössään elintärkeä [Bennet, 1995]. Perinnejärjestelmille on tunnusomaista korkea käyttöikä, sillä niitä käytetään sellaisissa kohteissa, joissa järjestelmän uusiminen voi olla liian riskialtista tai kallista, kuten lentoliikenteessä.

Integraation perusedellytys kuitenkin on, että kahden järjestelmän välillä on edes jonkinlaiset rajapinnat, joita voidaan hyödyntää. Tämän lisäksi järjestelmien välillä tarvitaan jokin fyysinen tiedonsiirtokanava, kuten tietoverkko tai internet [Tähtinen, 2005]. Harva järjestelmä on kuitenkaan niin suljettu, että integraatio olisi teknillisesti täysin mahdotonta.

Voidaankin sanoa, että integraation päävaatimus on järjestelmien yhteentoimivuus. Käsitteenä yhteentoimivuus voidaan jakaa vielä pienempiin osakokonaisuuksiin, jotka yhdessä muodostavat kokonaissuunnitelman eli kokonaisarkkitehtuurin.

Saboniha ja muut [2012] jakavat tietojärjestelmien yhteentoimivuuden seitsemään osatekijään, jotka muodostavat organisaatorakenteen kanssa erikokoisia tasoja. Teknologisella näkökulmalla tarkoitetaan integraation teknillisiä lähtökohtia, joka voivat olla tiedon välitystä, tietoturvallisuutta ja tiedon tallennusta. Syntaktisella yhteentoimivuudella tarkoitetaan tiedon siirtoa järjestelmien välillä, kun taas rakenteellinen yhteentoimivuus koskee yleisesti hyväksyttyjä malleja ja käsitteitä, jotka mahdollistavat tiedonsiirron. Semanttisen yhteentoimivuuden avulla välitettävästä tiedosta tulee integraation osapuolille ymmärrettävää ja mielekästä, kun taas operatiivisella yhteentoimivuudella tarkoitetaan sitä, miten hallinnollista ja tilastollista tietoa pitäisi tulkita. Organisaation yhteentoimivuudella tarkoitetaan suurempaa kokonaiskuvaa erilaisista rooleista ja käytännöistä, siitä miten integroitavaa tietoa käytetään organisaation sisällä.

Edellä mainittujen asioiden yhdistämistä varten on vuosien saatossa kehitetty lukuisia erilaisia lähestymistapoja ja integraatiomalleja. Näiden ideana on kerätä ja dokumentoida yleiskäyttöisiä malleja, joita voitaisiin soveltaa kulloisessakin käyttökontekstissa.

5. Yleisimmät integraatiotavat ja -mallit

Integroinnin päämääränä on siis lyhykäisyydessään hajautuneen tiedon yhdistäminen ja eteenpäin jakaminen. Asia ei kuitenkaan ole näin yksinkertainen, sillä on olemassa kymmeniä erilaisia integraatiomalleja ja tiedon jakelutapoja, joilla integraatio voidaan toteuttaa.

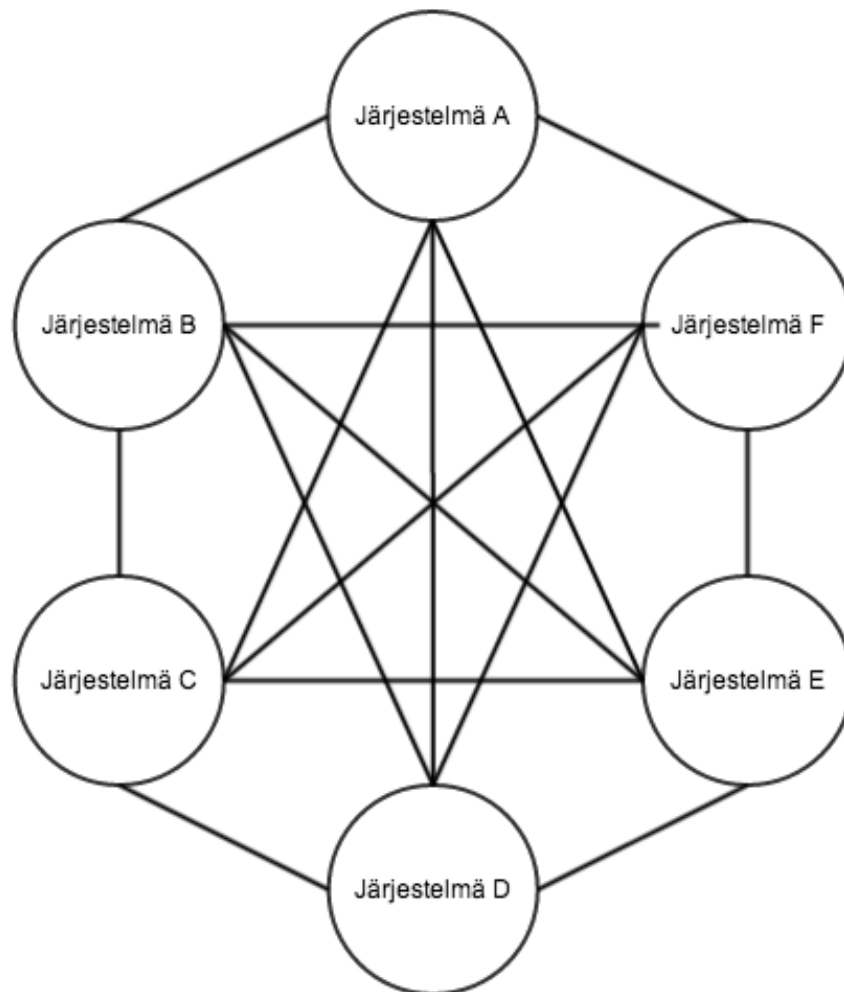
Teknisesti tiedonsiirto voidaan toteuttaa neljällä erilaisella tavalla: tiedoston siirrolla, jaetulla tietokannalla, etäproseduurikutsuin (Remote Procedure Call, RPC) tai sanomien avulla [Josuttis, 2007]. Jokaisella toteutustavalla on omat hyvät ja huonot puolensa, mutta viime vuosina tietojärjestelmäintegroinneissa on suosittu sanomapohjaisia ratkaisuja. Sanomien etuja ovat muun muassa mahdollisuus etäkommunikointiin, tiedonsiirto, luotettavuus ja helpompi integroitavuus muihin järjestelmiin.

5.1. Point-to-point

Yleensä pienet integraatioprojektit käynnistyvät siitä tarpeesta, että kaksi järjestelmää täytyy saada kommunikoimaan keskenään. Tähän tarpeeseen on syntynyt yksinkertaisin integraatiomalli eli Point-to-point- tai Siltayhteistoiminta-malli. Tässä mallissa järjestelmäintegraatio toteutetaan siten, että tieto siirretään ”sillan” kautta suoraan järjestelmästä toiseen. Tämän ansiosta yksittäisten integraatioiden toteuttaminen on nopeaa, helppoa ja suoraviivaista [Josuttis, 2007].

Toisaalta Point-to-point -integraation yksinkertaisuudessa piilee myös mallin heikkous, sillä erilaisia järjestelmäliittymiä on paljon ja niiden hallinnointi vaikeaa. Jos organisaatio hankkii uuden tietojärjestelmän, pitää uudesta järjestelmästä toteuttaa integraatiot kaikkiin jo olemassa oleviin järjestelmiin, eli linkkien määrä kasvaa neliöllisesti

käytettävien järjestelmien lukumäärän nähden. Tämän vuoksi Point-to-point -integraatiomalli onkin saanut nimityksen integraatiospagetti.



Kuva 3. Point-to-point -integraatiomalli

Jokaiseen liittymään pitää myös toteuttaa tiedonsiirron automatisointi. Jos järjestelmät eroavat huomattavasti toisistaan, jokaista yhteyttä varten tarvitaan erillisiä koodinmuuntimia, jotta järjestelmien väliset viestit pystymään kääntämään kohdejärjestelmän ymmärtämään muotoon [Mykkänen *et al.*, 2003]. Tämä koskee myös olemassa olevien järjestelmien päivitystä, jolloin liittymien muutos täytyy kartoittaa aina uudelleen. Tämän seurauksena integraatioista tulee vaikeasti ylläpidettäviä ja mahdollisten virheiden löytyminen vaikeutuu.

Point-to-point integraatiolle onkin tyypillistä, että niitä rakennetaan ainoastaan tarpeen vaatiessa, minkä seurauksena varsinainen integraatioarkkitehtuuri muovautuu hyvin pitkällä aikajänteellä. Vuosien saatossa järjestelmätoimittajat ja tekniikat voivat vaihtua, joten teknisten dokumentaatioiden paikkansapitävyys on hyvin keskeisessä roolissa integraatioarkkitehtuurin hallinnassa. Point-to-point -malli ei myöskään ole skaalautuva, joten yrityksen tiedonsiirtomäärien kasvaessa integraatiomallia joudutaan arvioimaan uudelleen.

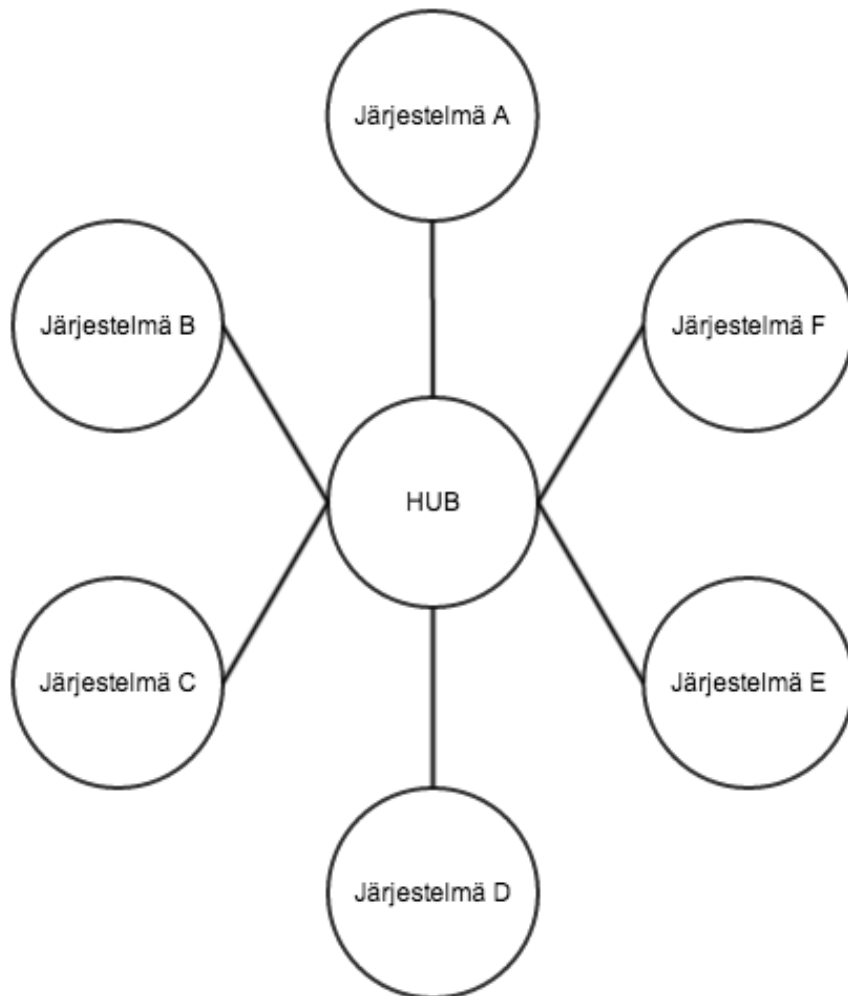
5.2. Hub-and-spoke

Hub-and-Spoke, eli perinteisemmin puhelinkeskusmalli oli vuosituhannen vaihteessa hyvin suosittu integraatoratkaisu. Tässä integraatiomallissa kaikki integroivat järjestelmät kytkeytyvät keskitettyyn integraatoratkaisuun, jonka vastuulla on viestien muunnos ja välttäminen kytkettyjen järjestelmien välillä. Keskitetyn ratkaisun ansiosta viestien välitystä ja muunnosta voidaan ohjata hallitusti, jolloin arkkitehtuurin hahmotus on helpompaa Point-to-point -malliin verrattuna [Bussler, 2010].

Uusien järjestelmien käyttöönotto, ja vanhojen järjestelmien päivitys on myös helpompaa, sillä Hub-and-spoke -mallissa toteutettavien liittymien määrä on suoraan verrannollinen olemassa olevien järjestelmien lukumäärään. Uuden järjestelmän käyttöönotossakin riittää ainoastaan se, että toteutetaan liittymä keskusjärjestelmään.

Arkkitehtuurissa ei olla niin riippuvaisia järjestelmätoimittajista ja näiden teknisistä dokumentaatioista, koska uusien järjestelmien ei tarvitse liittyä jo mahdollisesti vanhentuneihin perinnejärjestelmiin, joiden tekninen dokumentaatio voi olla puutteellista. Keskitetyn mallin etuina on myös tietoverkon ja informaatiovirtojen valvonta ja kontrollointi [Tähtinen, 2005]. Kaikkien järjestelmien ollessa kytkettyinä yhteen pisteeseen, voidaan hallita sitä, millaiset viestit tavoittavat halutun kohdejärjestelmän ja

kenellä on oikeus päästä käyttämään muiden järjestelmien resursseja. Hub-and-spoke -mallista on syntynyt vuosien aikana useita erilaisia variantteja, mm. DCOM, CORBA ja UDDI [Trowbridge *et al.*, 2012].



Kuva 4. Hub-and-spoke -integraatiomalli

Hub-and-spoke malli ei ole kuitenkaan täydellinen, sillä ongelmatilanteissa se on jo rakenteensa puolesta hyvin haavoittuva. Keskitetyn mallin keskusjärjestelmää kutsutaankin niin sanotuksi single point of failure -pisteeksi, eli tämä yksi piste rikkoutuessaan voi lopettaa koko yrityksen tiedonsiirron ja liiketoimintakriittisten järjestelmien toiminnan [Hohpe, 2012]. Näin ollen on ensiarvoisen tärkeää, että

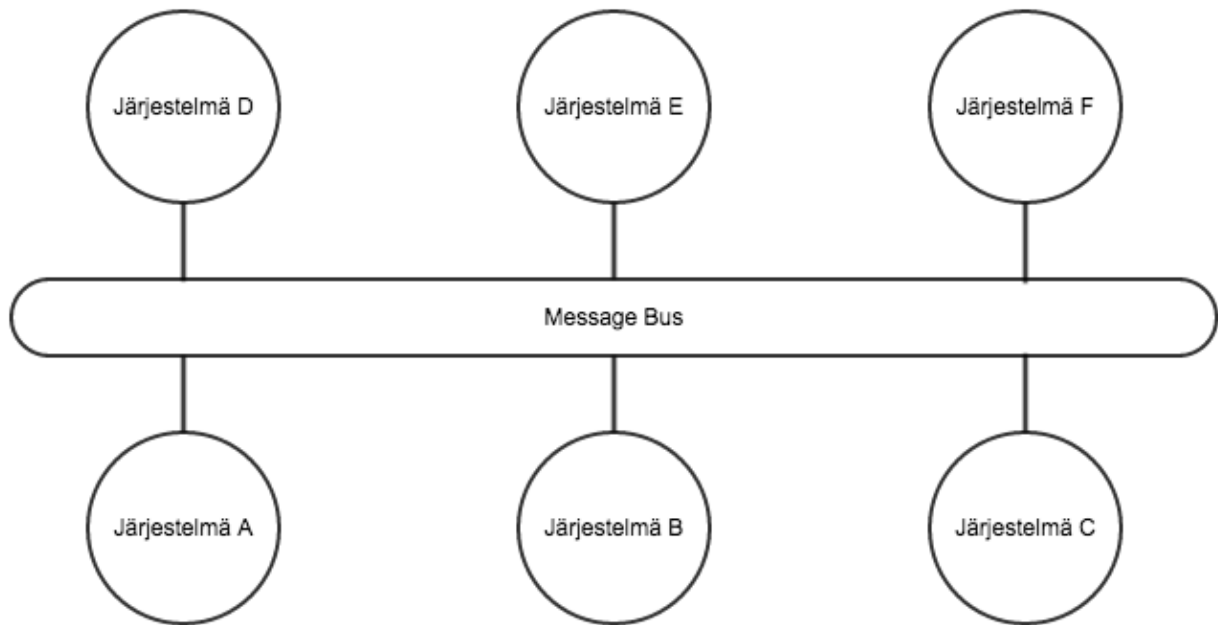
keskusjärjestelmä toimii luotettavasti ja on vikasietoinen. Kriittisten järjestelmien vikasietoisuutta voidaan kehittää kahdentamalla järjestelmiä ja kehittämällä valvontajärjestelmiä [Tähtinen, 2005]. Tämän lisäksi Hub-and-spoke mallin ongelmana on skaalautuvuus, sillä kaikki tiedonsiirtoliikenne tapahtuu yhden pisteen kautta.

Keskitetyn mallin heikkoutena on myös se, että tiedon lähettäjä ja vastaanottaja eivät ole toisistaan tietoisia. Siksi tapauksessa, jossa kaksi lähdejärjestelmää lähettää ristiriitaista tietoa, ei kohdejärjestelmän ole mahdollista päätellä, kumpi tieto on relevanttia. Tämän lisäksi Hub-and-spoke -järjestelmän käyttöönottokulut ovat suuremmat esimerkiksi Point-to-point -integraatioarkkitehtuuriin verrattuna, sillä tarvitaan yksi ylimääräinen liiketoimintakriittinen järjestelmä.

5.3. Message Bus

Message Bus -integraatiomalli on esitellyistä integraatiomalleista uusin, ja se on syntynyt tarpeesta paikata aikaisemmin esitettyjen integraatiomallien puutteet. Message Bus -mallin rakenne perustuu tietotekniikassa yleisesti käytettyyn väyläperiaatteeseen. Tässä ajatusmallissa järjestelmien kytkeminen ja poistaminen itse väylään on mahdollisimman helppoa ilman, että tällä on vaikutusta muihin siihen kuuluviin järjestelmiin.

Väylään kytkeytyneet järjestelmät kommunikoivat keskenään väylän kautta, jolloin väylän vastuulla on viestien eteenpäin välittäminen. Näin ollen viestin kulku Message Bus -mallissa ei ole determinististä [Trowbridge *et al.*, 2012]. Väylälle voidaan rakentaa erilaisia toteutuksia siitä, miten viestin lähetys käyttäytyy ja missä järjestyksessä viestit otetaan vastaan. Kohdejärjestelmät saavat oikeanlaista dataa, vaikka lähdejärjestelmissä tietolähteet eivät olisikaan yhteneväiset.



Kuva 5. Message Bus -integraatiomalli

Rakenteeltaan väylä on hajautettu, jolloin vältetään Hub-and-spoke mallin suorituskyky- ja skaalautuvuusongelmat. Hajautetulla tekniikalla tarkoitetaan tässä yhteydessä sitä, että tarvittavat koodinmuunnokset ja integraatiomootorit toteutetaan erikseen jokaiseen liittymään, mitkä taas toimivat samalla alustalla kuin itse väylä. Näin saavutetaan parempi skaalautuvuus ja teknologian yhteentoimivuus. Väylän avulla voidaan myös vähentää järjestelmien välistä riippuvuutta ja mahdollistaa toimittajariippuvuutta. Integraatioiden toteutuskulut ja työmäärät ovat huomattavasti pienemmät, kuin esimerkiksi Point-to-point -mallissa [Trowbridge *et al.*, 2012].

Message Bus -termi saatetaan sotkea käsitteeseen ESB, Enterprise Service Bus. Vaikka ESB on tekniseltä rakenteeltaan Message Bus -mallin kaltainen, on se paljon muutakin kuin pelkästään väylä viestin välitykseen. Palvelukeskeisestä arkkitehtuurista ja ESB:tä tarkemmin vielä lisää työn seuraavissa kappaleissa.

6. Palvelukeskeinen arkkitehtuuri

Palvelukeskeinen arkkitehtuuri, Service-Oriented Architecture tai SOA ovat käsitteitä, joita on käytetty viime vuosikymmenen ajan hieman leväperäisesti, tarkoittaen hyvinkin erilaisia asioita. Palvelukeskeinen arkkitehtuuri voidaan nähdä kuuluvaksi osana palvelukeskeisen tietotekniikan käsitteeseen, vaikka useasti näitä käytetäänkin synonyymeina. Erl:n [2008] mukaan palvelukeskeisen tietotekniikan käsitteeseen sisältyy kuitenkin myös paljon muuta, kuten palvelukeskeisyyden käsite, palvelukeskeinen ratkaisulogiikka, palvelut, palveluiden yhdistäminen ja palvelutarjooma.

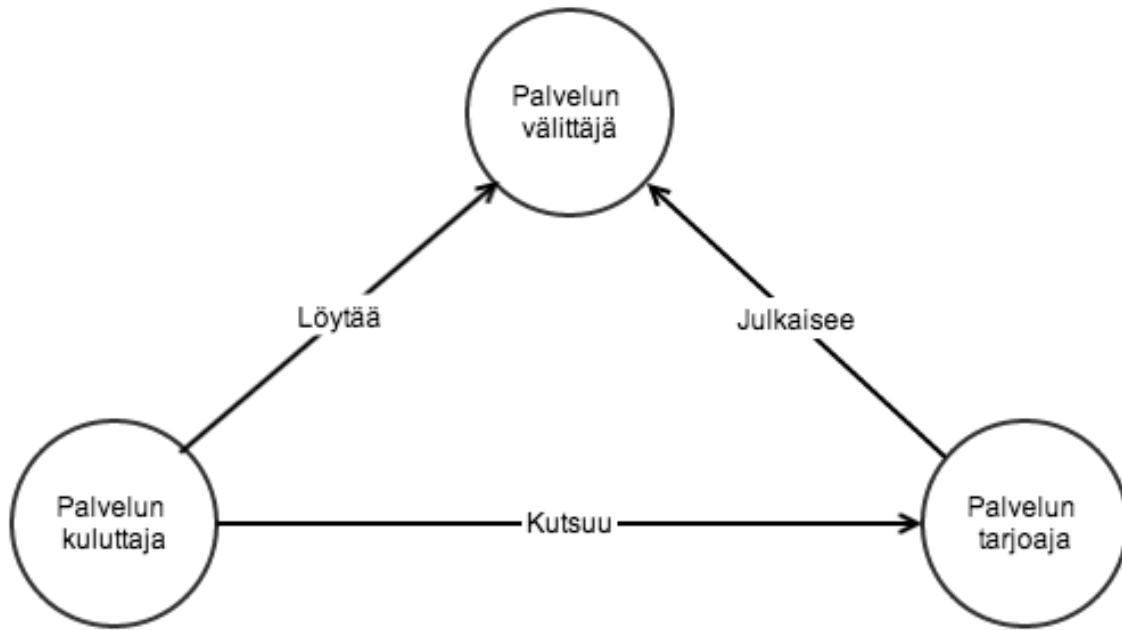
Palvelukeskeistä arkkitehtuuria ei myöskään pidä sekoittaa kokonaisarkkitehtuurin (Enterprise Architecture) käsitteeseen, jolla taas tarkoitetaan välinettä liiketoiminnan ja ICT:n yhtäaikaiseen kehittämiseen ja hallintaan. Kokonaisarkkitehtuurilla siis yritetään suunnitella organisaation toimintaa ja kuvata mahdollisia tietojärjestelmätarpeita [Siltanen, 2004] Voidaankin sanoa, että kokonaisarkkitehtuurin päämääränä on keskittyä siihen mihin organisaatio strategisesti haluaa mennä, kun taas palvelukeskeisen arkkitehtuurin vastuulla on se miten tavoitteet saavutetaan.

Tämän kappaleen keskiössä tulee kuitenkin olemaan palvelukeskeinen arkkitehtuuri. Tätä voidaan kuvata paradigmana, jonka avulla havainnoidaan ja hallitaan liiketoimintaprosesseja. Nämä prosessit voivat olla levittäytyneet laajalle hajautettujen järjestelmien ansiosta. Näin ollen SOA ei ole yksittäinen tuote, jonka organisaatio voisi ostaa käyttöönsä, vaan asenne tai ajattelutapa, johon organisaation pitäisi pyrkiä [Josuttis, 2007].

6.1. Arkkitehtuurin pääpiirteet

Palvelukeskeisessä arkkitehtuurissa lähdetään siitä ajatuksesta, että ohjelmistot ja tietojärjestelmät tarjoavat erilaisia palveluita yrityksen kokonaisratkaisun rakentamiseksi. Tällä tarkoitetaan sitä, että järjestelmä ei ole ainoastaan yhden komponentin varassa vaan vastuu on jaettu eriytettyjen yksiköiden kesken, jotka yhdessä muodostavat isomman kokonaisuuden.

Yksinkertaistettuna esimerkkinä voidaan pitää tilannetta, jossa yrityksen henkilöstönhallintajärjestelmä tarjoaa palvelun, jonka kautta voidaan kysellä yrityksessä työskentelevien henkilöiden perustiedot. Tämä palvelu toteutetaan sovitulla rajapinnalla, joten palkanmaksujärjestelmä voi käydä kysymässä tarvitsemansa tiedot suoraan henkilöstönhallintajärjestelmältä. Järjestelmäarkkitehtuuri koostuu siis useista komponenteista, jotka osaavat hyödyntää toinen toistensa tarjoamia palveluita. Tämän ansiosta yksittäiset komponentit voivat erikoistua omaan erityisosaamiseensa ja järjestelmäkentästä tulee hallitumpi kokonaisuus. Edellä kuvattu tapa lisää myös järjestelmien joustavuutta, joka on yksi SOA -ajattelun keskeisistä päämääristä. SOA -ajattelun peruseriaate on kuvattu kaaviokuvassa 6.



Kuva 6. Palvelukeskeisen arkkitehtuurin toimintamalli

Periaatetasolla mallissa ei ole mitään uutta, sillä uudelleenkäytettävyys on ollut jo pitkään tavoitteena perinteisessä ohjelmistokehityksessä. Palvelukeskeinen arkkitehtuuri pyrkiikin toteuttamaan järjestelmäintegraation tapauksessa samat hyväksi todetut käytännöt, nyt vain ohjelmistokehitystä ylemmillä abstraktiotasoilla [Josuttis, 2007].

Toinen SOA:n keskeinen pääpiirre on tuki heterogeenisille IT-järjestelmille. Nykyään isoissa organisaatioissa käytetään hyvin monia tietoteknisiä sovelluksia, eikä aina ei voida olettaa, että kaikki tietojärjestelmät olisivat teknisiltä ratkaisuiltaan samanlaisia. Tämä ajattelutapa korostuu tulevaisuudessa, kun pilvi- ja verkkosovelluspalvelut yleistyvät. Siksi SOA -ajattelulle on keskeistä, että se tukee myös heterogeenisiä ympäristöjä [Josuttis, 2007].

6.2. Palvelu

Jotta voimme puhua tarkemmin palvelukeskeisestä arkkitehtuurista, on määriteltävä mitä tarkoitetaan palvelulla. Tässä yhteydessä palvelulla tarkoitetaan fyysisesti itsenäisiä

loogisia yksikköjä, jotka eivät kuitenkaan saa olla eristyksissä muista yksiköistä. Näiden loogisten yksiköiden tai komponenttien täytyy toteuttaa myös jotain tiettyjä yhteisiä standardeja, jotta ne voisivat kommunikoida keskenään [Erl, 2008]. Pääsääntöisesti palvelut ovat kuitenkin löyhästi sidoksissa toisiinsa, komposiittisia, tilattomia, uudelleenkäytettäviä ja abstraktiotaso on suunniteltu siten, että palvelulogiikkaa ei näy palvelusta ulospäin. Voidaankin sanoa, että SOA -arkkitehtuurissa palvelu on IT-maailman kuvaus jostain liiketoiminnan rajatusta toiminnallisuudesta.

Palveluita ja näiden luokitteluja on useita, mutta Erl [2008] jakaa teoksessaan palvelut kolmeen eri ryhmään: entiteetteihin, tehtäviin ja työkaluihin. Entiteettipalvelut sisältävät tyypillisimmillään tietoa organisaation perustiedoista, kuten asiakkaista, työntekijöistä ja laskuista. Entiteettipalveluihin kuuluvat perinteiset tietotekniikan luku ja kirjoitusoperaatiot, eli tietojen syöttö, luku, päivitys ja poisto.

Työkalut taas tarjoavat nimensä mukaisesti uudelleen käytettävinä palveluina erilaisia yleishyödyllisiä työkaluja, kuten virheidenkäsittely ja tapahtumahistoria. Tehtäväpalvelut ovat monitahoisempi käsite, sillä näiden vastuulle tulee muun muassa muiden palveluiden organisointi. Näin ollen tehtävät koostuvat useista muista palvelutyypeistä ja niiden vastuulle tulee myös liiketoimintasäännöistä huolehtiminen [Erl, 2008].

Mitä hyötyä tällaisten palveluiden käytöstä sitten on? Nykyaikana tietotekniikka ja sen tarjoamat mahdollisuudet muuttuvat alati kiihtyvällä vauhdilla, minkä seurauksena kukaan ei voi tarkasti ennustaa millainen tulevaisuus tulee olemaan. Palveluiden ja komposiittisen rakenteen avulla järjestelmät voivat kuitenkin yhä paremmin reagoida nopeaan muutokseen. Muutoksen yhteydessä voi riittää ainoastaan yksittäisen palvelun päivitys, eikä muuhun sovellukseen tarvitse koskea.

6.3. Yhteentoimivuus

Palveluiden lisäksi SOA -arkkitehtuurin yksi keskeisimmistä käsitteistä on yhteentoimivuus [Josuttis, 2007]. Yhteentoimivuudella tarkoitetaan sitä, että uusien järjestelmien pitää olla mahdollisimman helposti kytkettävissä ja käytettävissä jo olemassa olevien järjestelmien kanssa. Yhteentoimivuutta voidaan parantaa muun muassa sillä, että järjestelmän tarjoamat palvelut ovat kaikkien muidenkin järjestelmien käytettävissä ja siten, että järjestelmät käyttävät yleisesti tunnettuja rajapintastandardeja. Tämäkään ajattelutapa ei ole ainoastaan SOA-ajattelutavalle tyypillistä, vaan se on ollut kantava ajatuksena muissakin aikaisemmin esitetyissä integraatiomalleissa, kuten esimerkiksi Hub-and-Spoke -mallissa.

6.4. Löyhät sidokset

Kolmas keskeinen käsite palvelukeskeisessä arkkitehtuurissa on järjestelmien väliset löyhät sidokset [Josuttis, 2007]. Löyhillä sidoksilla tarkoitetaan tässä sellaista tilaa, jossa palvelu hankkii tietoa toisista palveluista, kuitenkin siten, että kysyvä palvelu pysyy muista riippumattomana [Erl, 2008].

Jotta SOA -mallissa pystyttäisiin välttämään aikaisemmin kuvattujen integraatiotapojen heikkoudet, täytyy SOA:n olla joustava ja skaalautuva, mutta samalla hyvin vikasietoinen. Näihin päämääriin päästääkseen pitää järjestelmäarkkitehtuurin pyrkiä mahdollisimman löyhiin sidoksiin yksittäisten järjestelmien välillä, jotta vältettäisiin turhia riippuvuuksia [Erl, 2008]. Tällä tarkoitetaan sitä, että jos yksi järjestelmä vikaantuu, ei se välttämättä vaikuta muiden järjestelmien toimintaan. Löyhien sidosten myötä järjestelmäkenttään ei myöskään synny tarpeettomia keskittymiä, minkä ansiosta välttyään turhilta pullonkauloilta ja järjestelmästä tulee skaalatuvampi.

6.5. Palveluväylä

ESB-kirjainlyhennelmällä tarkoitetaan jo aikaisemmin esiteltyä käsitettä Enterprise Service Bus (jatkossa palveluväylä tai väylä). Palveluväylän päätarkoituksena on toteuttaa edellä kuvattu yhteentoimivuuden vaatimus [Josuttis, 2007]. Teknisesti palveluväylä muodostuu kokoelmista erilaisia tekniikoita, joiden avulla voidaan toteuttaa SOA -arkkitehtuurin mukainen viestinvälitys. Näin ollen palveluväylää voidaan pitää Hub-and-Spoke -mallin kehittyneempänä versiona.

6.5.1. Palveluväylän tehtävät

Palveluväylä on siis SOA -arkkitehtuurin viestinvälityksen ydin, jonka avulla organisaation tietojärjestelmät voivat tarjota omia palveluitaan muiden ohjelmistojen hyödynnettäviksi. Palveluväylälle voidaan myös nimensä mukaisesti toteuttaa erinäisiä palveluita. Jotta heterogeeninen IT-järjestelmäkenttä saataisiin yhdistettyä väylään, täytyy väylän olla teknisesti helposti yhdistettävissä. Yhdistettävyyden onkin yksi väylän, ja samalla koko yhteentoimivuuden käsitteen, keskeisimpiä osa-alueita [Josuttis, 2007].

Yhdistettävyyden lisäksi palveluväylän pitää tarjota mahdollisuus tiedon muunnokseen, jotta erilaiset järjestelmäalustat ja ohjelmistokielet voivat kommunikoida keskenään. Yleensä tiedon muunnokset toteutetaan sisäisenä palveluna, minkä ansiosta yksittäiseen palveluväylään liittyneen sovelluksen sijaintiin ei tarvitse ottaa kantaa. Muunnosta varten toteutetaan hyvin usein oma, yhteinen tietöformaatti, kanoninen malli (Canonical model) [Hohpe, 2012]. Kanonista mallia esitellään tarkemmin työn myöhemmässä vaiheessa.

Yhdistettävyyden ja tiedon muunnoksen lisäksi palveluväylän keskeisenä tehtävä on toteuttaa tiedon reititys. Jotta väylään liittyneet järjestelmät voisivat hyödyntää toisiaan, pitää väylän pystyä välittämään palvelun kutsujan viesti palvelua tarjoavalle

järjestelmälle, ja toisaalta myös mahdollinen vastaus takaisin. Tyypillisesti palveluväylällä liikkuvat viestit ovat XML-pohjaisia sanomia.

6.5.2. Palveluväylän tekniset toteutustavat

Vaikka tässä työssä on puhuttu paljon väylästä yhtenä tiettyä integraatiotyökaluna, pitää kuitenkin muistaa, että kyseessä on ainoastaan fundamentaalinen malli, jonka reaali maailman toteutukset voivat erota paljonkin toisistaan. Palveluväylä-käsitettä ei ole teknisesti sidottu mihinkään ohjelmointikieleen tai toteutusalueeseen, mutta myös se miten väylään kiinnitettävät järjestelmät liittyvät ja näkevät toisensa, voi vaihdella tapaus- ja teknologiakohtaisesti.

Vanhanaikaisin, ja samalla myös vähiten SOA -arkkitehtuurin mukainen tapa toteuttaa väylän tekninen ratkaisu on point-to-point -integraatioiden piilotus väylän sisälle [Josuttis, 2007]. Tässä tapauksessa väylän palveluiden kutsu toteutetaan niin, että palvelun kutsujan täytyy tietää palvelun tarjoajan suorat ”yhteystiedot”. Tämän toteutustavan heikkoudet ovat enimmäkseen samat, kuin point-to-point -integraatiomallin.

Edistyneempi väylän toteutustapa toimii niin, että kutsuva järjestelmä tuntee palveluntarjoajan esimerkiksi yhteisesti sovitulla tunnuksella, ja varsinainen tiedonvälitys jää väylän omaksi tekniseksi ratkaisuksi. Tämän niin kutsutun välitysmallin etuna on se, että esimerkiksi kutsuttavan järjestelmän osoitteen muutokset eivät vaikuta varsinaiseen palvelukutsuun [Josuttis, 2007]. Tämän lisäksi palveluväylälle voidaan toteuttaa liittymäkohtaisia priorisointeja, mahdollista kuorman tasausta ja parempaa virheiden hallintaa.

Välitysmallin lisäksi SOA -ajattelutavan mukaisia löyhiä liitoksia pystytään toteuttamaan lisäämällä välityspalvelimia väylätoteutukseen. Tässä tapauksessa palvelun kutsuja kutsuu välityspalvelinta kuten kutsuisi varsinaista palveluntarjoajaa point-to-point -

integraatiomallissa. Välityspalvelimen tehtävänä on tämän jälkeen saattaa viesti oikeaan osoitteeseen. Välityspalvelimen yhteyteen on myös mahdollista toteuttaa erilaisia kuormantasaajia.

6.5.3. Palveluväylän hyödyt

Kuten aikaisemmista kappaleista on käynyt ilmi, palveluväylä ei ole yksittäinen, massiivinen ohjelmisto, vaan se koostuu useista pienistä komponenteista. Näin arkkitehtuurista tulee skaalautuvampi, ja samalla vältetään mahdollinen single point of failure -tilanne, joka oli heikkoutena esimerkiksi Hub-and-Spoke -integraatiomallissa. Palveluväylän komponentteihin voidaan myös toteuttaa monimuotoisia palveluita, joilla tuotetaan käyttäjille lisäarvoa.

Koska palveluväylästä muodostuu helposti organisaation IT -arkkitehtuurin sydän, niin yksi keskeisistä eduista on viestien keskitetty tallennus ja monitorointi. Onkin loogista ottaa talteen järjestelmien väliset viestit siinä pisteessä, jonka kautta kaikki viestit kulkevat [Tähtinen, 2005].

Lisäksi palveluväylän yhteyteen voidaan toteuttaa reititystä, joka on älykkäämpää kuin perinteinen viestinvälitys. Organisaatiolla voi olla käyttötarpeita, jossa tietyt viestit tai tiedon lähdejärjestelmät ovat prioriteetiltaan korkeammalla kuin muut tietojärjestelmät. Reitityksen yhteyteen voidaan toteuttaa myös erilaisia tiedonpäättelysäännöstöjä, joiden avulla väylällä kulkevat viestit saadaan samanlaiseen formaattiin [Josuttis, 2007]. Esimerkkinä tästä on tekstikenttien kirjoitusasun yhtenäistäminen.

Palveluväylän yhteyteen on myös mahdollista rakentaa BAM-toteutus, Business Activity Monitoring. BAM:n ydinajatuksena on, että väylän kautta kulkevien viestien perusteella liiketoiminta pystyy tekemään lähes reaaliaikaisia päätöksiä liiketoiminnan edistämiseksi. BAM:n tarjoamia mahdollisuuksia tarkastellaan tarkemmin kappaleessa 5.7.

6.6. Kanoninen malli

Eri tietojärjestelmiä integroitaessa lähdejärjestelmät voivat poiketa toisistaan hyvinkin paljon, minkä takia käytössä oleva tieto ei ole yhteismitallisessa muodossa. Tämä on hyvin loogista, sillä tietoa lähettävä järjestelmä viestittää tietoa pääsääntöisesti kyseisen järjestelmän tuntemaan tietomallin muodossa [Hohpe, 2012]. Näin ollen tätä tietoa joudutaan mahdollisesti muuntamaan erilaisilla tiedon muuntimilla, jotta palveluväylään kiinnitetyt järjestelmät voisivat kommunikoida keskenään.

Muuntimet voidaan tehdä tiedon lähde- ja kohdejärjestelmäkohtaisesti, mutta jos yhdistettäviä järjestelmiä on paljon, kasvaa muuntimien määrä eksponentiaalisesti [Schmidt and Lyle, 2010]. Muuntimien määrää kasvattaa myös se, että palvelut voivat lähettää useita erilaisia viestejä, joille pitää taas toteuttaa omat käännoöksensä. Jos palveluiden tietosisältö muuttuu, tarkoittaa se samalla myös kaikkien muuntimien päivytystä, joten niiden ylläpito on hyvin aikaa vievää.

Ongelmat ovat siis hyvin pitkälle samoja kuin point-to-point -integraatiomallissa. Tämän vuoksi onkin kehitetty suunnitteluperiaate, kanoninen tietomalli, jota käytetään yleisesti osana palvelukeskeistä arkkitehtuuria ja palveluväylää. Arkkitehtuurin ideana on, että luodaan organisaation hyväksymä tietomalli, jonka kaikki palveluväylään kiinnittyvät palvelut toteuttavat [Hohpe, 2012].

Kanoninen tietomalli on siis muista sovelluksista riippumaton tapa muuntaa tieto yhteismitalliseen muotoon, ja samalla vähentää riippuvuuksia järjestelmiä integroitaessa. Esimerkiksi kun uusi järjestelmä otetaan käyttöön tai vanha järjestelmä korvataan uudella, tarvitsee tämän toteuttaa vain kanoninen tietomalli, ja siten tieto on muiden järjestelmien käytettävissä. Tämä edistää palveluiden yleiskäyttöisyyttä ja helpottaa olemassa olevien tiedonmuuntimien hallintaa. Yhteisen tietomallin etuna on myös se, että organisaation liiketoiminnan edustajien on mahdollista keskustella integraatioiden sisällöstä, kun

käytössä on organisaation laajuinen ja uudelleenkäytettävä tietomalli [Schmidt and Lyle, 2010].

Kanonisen mallin heikkoutena on sen mahdollinen raskaus, jos integraatioiden lukumäärä on kohtalaisen pieni tai kyseinen käyttötapaus on hyvin aikakriittinen. Tämä johtuu siitä, että jokaiseen saapuvaan viestiin pitää toteuttaa kaksi muunnosta, lähdejärjestelmästä yhteiseen malliin ja yhteisestä mallista kohdejärjestelmään. Onkin tapauskohtaista, milloin kanoninen tietomalli kannattaa ottaa osaksi palvelukeskeistä arkkitehtuuria.

6.7. BAM

Nykyaikaisten yritysten haasteena on ollut se, miten reaaliaikaisesti voitaisiin valvoa ja optimoida liiketoimintaprosesseja ja samanaikaisesti kehittää ja ylläpitää liiketoimintaa. Kaiken toiminnan ollessa sähköistettyä ja tietojärjestelmien pyörittämää, saatavilla olevan tiedon määrä muuttuu valtavaksi, eikä sitä ole enää järkevää tai mahdollista ihmisvoimin analysoida.

Liiketoimintaprosesseissa voi myös tapahtua mahdollisia virheitä ja erikoistilanteita. Olisikin suotavaa, että nämä tilanteet havaittaisiin mahdollisimman nopeasti, jotta niihin voitaisiin reagoida oikein. Tiedon täytyy kuitenkin jollain tasolla olla jalostettuna, jotta organisaation työntekijä löytää siitä itselleen oleellisen informaation. [Engel *et al.*, 2013.]

Tiedon saannin ja jakelun keskiössä on se työkalu, jonka kautta kulkee mahdollisimman paljon organisaation tietoa. Nykyaikaisessa, palvelun laatua parantavassa integraatioarkkitehtuurissa palveluväylä on hyvin luonnollinen paikka tälle tiedon jakelulle. Palveluväylältä tietoa voidaan siirtää perinteisen mallien mukaisesti tietovarastoon, josta se on organisaation raportointityökalujen käytettävissä. Jotta yrityksen prosessien hallinnasta tulisi mahdollisimman ketterää, voidaan palveluväylän yli kulkevia tapahtumia monitoroida lähes reaaliaikaisesti. [Tähtinen, 2005.]

Tätä toimintaa varten on kehitetty tuotteita, jotka ovat saaneet nimen Business activity monitoring (BAM). Käsitteen BAM on ensimmäisen kerran esittänyt Gartner-tutkimuslaitos 2000-luvun alussa. BAM-tuotteiden tarkoituksena on, että yritys saa reaaliaikaista tietoa liiketoimintaprosessiensa tilanteesta ja tapahtumista. Erilaisia BAM-ratkaisuja on lukuisia, mutta yhteistä näille on se, että niihin voidaan määritellä kulloiseenkin tarpeeseen sopivia kojelautoja (dashboard), jotka sisältävät tarvittavat avaintulosmittarit. Integraatiokerroksen läpi tulevista viesteistä voidaan tehdä erilaisia laskelmia ja analyysyjä, joiden tulokset voidaan visualisoida liiketoiminnan ymmärtämässä muodossa. Näin liiketoiminnan johdolla on lähes reaaliaikainen käsitys prosessien tilasta, mikä mahdollistaa nopean päätöksenteon ja mahdollisten ongelmakohtien tunnistamisen. [Kang and Han, 2008.]

7. Liiketoimintaprosessit

Liiketoimintaprosesseilla tarkoitetaan toisiinsa liittyneiden toimintojen muodostamaa kokonaisuutta. Prosessi alkaa asiakkaan tarpeesta ja päättyy tarpeen tyydyttämiseen [Tähtinen, 2005]. Tyypillisiä liiketoimintaprosessi käsitteen alle kuuluvia osa-alueita voivat olla esimerkiksi myynti-, tuotanto-, markkinointi- tai taloushallinnon prosessit.

Yritysten välisillä (jatkossa business-to-business, B2B) liiketoimintaprosesseilla tarkoitetaan prosessien sekvenssiä, jotka tapahtuvat joko yrityksen sisällä tai sen ulkopuolella. Voidaankin siis sanoa, että yritysten välisten liiketoimintaprosessien tarkoituksena on suorittaa yritysten välinen vuorovaikutus. Tietotekniikan aikakaudella tämä B2B -vuorovaikutus on sähköistynyt hyödyntäen erilaisia sähköisiä dokumenttistandardeja. Näin pyritään automatisoimaan ja tehostamaan liiketoimintaprosesseja. [Bussler, 2010.]

7.1. Liiketoimintaprosessien viestintästandardit

Tietoverkkojen kehittymisen alkuvaiheessa järjestelmien välinen kommunikaatio toteutettiin varsin räätälöidyillä rajapinnoilla ja sanomatekniikoilla. Tekniikoiden käyttötarkoituksena oli palvella rajattua yritystä, yritysrypästä tai toimialaa [Tähtinen, 2005]. 1960-luvun jälkeen erilaiset viestintätavat ja standardit alkoivat kuitenkin kehittyä rinta rinnan yleisen tietoteknisen kehityksen kanssa.

Yksi ensimmäisistä standardeista on ollut Electronic data interchange (jatkossa EDI), joka kehittyi 1970-luvun loppupuolella tarjoten räätälöityjen tekniikoiden rinnalle yleiskäyttöisemmän kommunikaatitavan. EDI-stantardin lähtökohtana on ollut muuntaa perinteisten paperidokumenttien sisältö sähköiseen muotoon, joten sanomien sisältö mukailee myös tätä. Rakenteeltaan EDI-sanomat ovat riviperustaista, rakenteista tietoa,

joiden erotteluun on käytetty erotinmerkkiä. EDI-standardia on yleisesti käytetty tiedonsiirtotapana erityisesti yritysten välisessä sähköisessä kaupankäynnissä, ja sitä varten on toteutettu kaksi pääasiallista standardia, ANSI ASC X12 ja EDIFACT [Bussler, 2010]. Useat suuret toimialat, kuten kemikaali- ja kaasuala, ovat kehittäneet myös omia tarkempia välitysformaattejaan, jotka ovat kuitenkin perustuneet EDI -standardiin.

EDI on kuitenkin nimensä mukaisesti vain yhteyskäytäntö, jonka avulla organisaatiot ovat voineet välittää esimerkiksi tilaustietoa sähköisesti eteenpäin. Yhteyskäytännön heikkoutena on sen kankeus sekä se, ettei EDI mahdollista monimutkaisten prosessien kuvausta [Bussler, 2010]. Tämä taas ei ole riittänyt uusiutuville yrityksille, jotka haluavat tehostaa liiketoimintaprosessejaan ja parantaa yritysten keskinäistä yhteenliittymistä. Tämän vuoksi EDI-sanomien rinnalle on vuosien aikana kehittynyt useita nykyaikaisempia standardeja, kuten ebXML ja UBL [Bussler, 2010]. Näiden määrittämisperheiden avulla yritykset voivat saavuttaa kehittyneempiä julkisia prosesseja ja mahdollisesti automatisoida osan sähköisistä toiminnoistaan.

7.2. Integraatioarkkitehtuuri osana liiketoimintaprosesseja

Mitä tekemistä liiketoimintaprosesseilla on sitten integraatioarkkitehtuurin kanssa? Tässä työssä on keskitytty käsittelemään organisaation sisäistä integraatiota, mutta B2B -kommunikaation kasvava tarve on johtanut siihen, että organisaation ulkoinen integraatio on aivan yhtä tärkeää kuin sisäinen integraatio. Ydinajatuksena on, että myös yrityksen omien tietojärjestelmien ja palomuurien takana on elämää, josta liiketoiminta voi ammentaa tietoa. Tästä on luontaisesti syntynyt tarve päästä tähän kullanarvoiseen tietoon käsiksi [Tähtinen, 2005].

Yritysten välinen verkostoituminen on tuonut organisaation liiketoimintaprosesseihin elementtejä, jotka tapahtuvat organisaation ulkopuolella. Tämä on vuorostaan asettanut integraatioarkkitehtuurille vaatimuksen toimia prosessin eri vaiheissa erilaisiin tehtäviin

erikoistuneiden tietojärjestelmien yhdistelijänä. Näin ollen järjestelmäintegraatio on hiljalleen muuntumassa tekniseltä tasolta lähemmäksi liiketoimintaprosessien hallintaa. [Tähtinen, 2005].

Integraatioarkkitehtuurin, viestintästandardien, organisaation sisäisten prosessien ja B2B - prosessien välistä suhdetta voidaan havainnollistaa perinteisten liiketoimintaprosessien, kuten myyntilaskupalvelun (OTC, Oder to cash) kautta. Myyntilaskupalvelu on kuvattu kuvassa 7.



Kuva 7. Order to Cash -prosessin yksinkertaistettu malli

B2B myynti alkaa yleensä siitä, että saatavilla olevien tuotteiden hinnasto toimitetaan sähköisessä muodossa asiakkaalle. Tämä tieto siirtyy EDIFACT -standardissa PRICAT - sanomassa. Hinnaston perusteella asiakas voi tuottaa ORDERS -sanoman, joka sisältää tilauksen tiedot. Tilaus lähetetään sähköisiä viestintäkanavia pitkin ja kun tilaus on vastaanotettu, myyjäorganisaatio voi kuitata tilauksen ORDRSP -viestillä. [UN/EDIFACT, 2014].

Tämän jälkeen tilaus voi edetä myyjäorganisaation omaan toiminnanohjausjärjestelmään, jossa tilausta ruvetaan käsittelemään sisäisten prosessien mukaisesti. Osana toiminnanohjausjärjestelmän prosesseja voidaan muodostaa DELFOR -sanoma, jonka integraatiokerros välittää sähköisiä kanavia pitkin asiakkaalle [UN/EDIFACT, 2014]. Tämä sanoma kertoo toiminnanohjausjärjestelmän antaman arvion tilauksen aikataulusta.

Tilauksen ollessa valmis kuljetukseen lähetetään EDIFACT -standardin mukainen DESADV- viesti, joka sisältää luettelon lähetetyistä tuotteista tai tuotteista, jotka ovat vallitsevan sopimuksen puitteissa valmiita lähtemään. Toimituksen jälkeen myyjäorganisaatio toimittaa laskun INVOIC -viestillä. [UN/EDIFACT, 2014].

Edellä kuvatun kaltaisessa B2B -prosessissa organisaation toiminnanohjausjärjestelmän ja integraatiokerroksen merkitys on suuri, sillä niiden avulla pystytään tuottamaan ja kommunikoimaan tietoa asiakkaan järjestelmiin, mutta samalla hallitsemaan myös organisaation sisäisiä prosesseja.

8. Tapaustutkimus - teollisuuden kunnossapitopalveluita tarjoava yritys

Tämän tapaustutkimuksen kohteeksi valikoitui järjestelmäintegraatioprojekti, jonka toteutustiimissä toimin yhtenä toteuttajana noin vuoden ajan. Projektin päämääränä oli tuottaa asiakkaalle integraatioarkkitehtuuri, jonka avulla asiakas voisi uudistaa omia liiketoimintaprosessejaan, ja samalla hyödyntää viimeisimpiä teknologisia sovelluksia. Projektin ansiosta yritys pystyisi tarjoamaan asiakkailleen kustannustehokkaampaa kokonaisresursointia, läpinäkyvyyttä ja parempaa raportointia.

8.1. Asiakkaan lähtötila

Tapaustutkimuksen kohteena olevan yrityksen lähtötila oli hyvin tyypillinen organisaatiolle, joka on päättänyt aloittaa massiivisen tietojärjestelmähankkeen. Yrityksen omistaja päätti uudistaa organisaation strategian ja sen pohjalta aloittaa merkittävän kehityshankkeen. Uudistuneen strategian avulla yritykseltä toivottiin reilua kasvua ja tuottavuuden nousua, minkä seurauksena tietojärjestelmiin kohdistuvia resursseja lisättiin huomattavasti.

Muutosta lähdettiin tavoittelemaan uudistamalla organisaation keskeiset liikeprosessit. Prosessit otettiin huolellisen tarkastelun alle, auottiin pienempiin osakokonaisuuksiin ja arvioitiin tukevatko ne uusia tavoitteita. Osa vanhoista liiketoimintaprosesseista uudistettiin kokonaan, sillä nähtiin, että ne olivat jo aikansa eläneet.

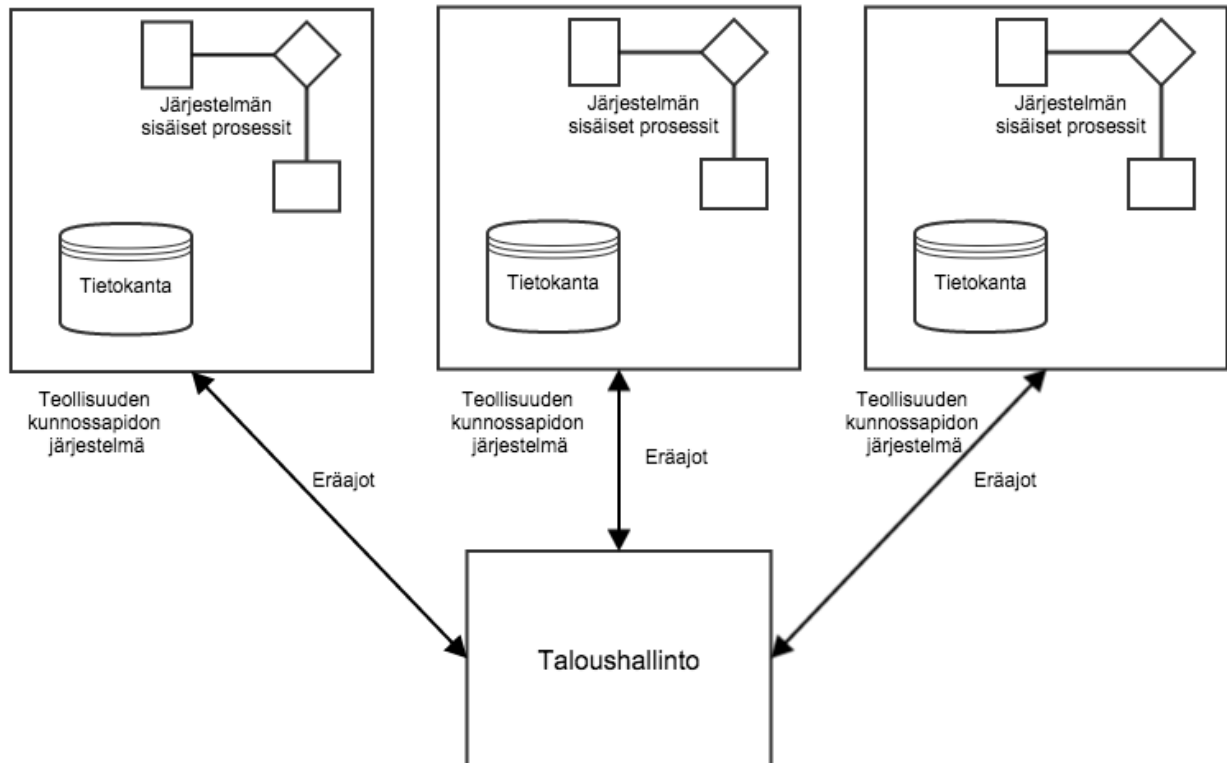
Tämän työvaiheen keskeisimpänä löytönä voidaan kuitenkin pitää sitä, kun huomattiin, että vanhat tietojärjestelmät eivät tue tarpeeksi työntekijän päivittäistä tekemistä. Aikaisemmin työntekijän täytyi itse mennä tiedon luokse, mutta uudistuneessa organisaatiossa haluttiin, että tieto on mahdollisimman lähellä työntekijän todellista arkea.

Näin pystyttäisiin saavuttamaan nopeampi reagointikyky, ja lisäämään työn tehokkuutta. Tarkastelun yhteydessä uudet ja uudistuneet prosessit dokumentoitiin huolellisesti liiketoiminnan näkökulmasta. Tällä oli erityisen tärkeä merkitys varsinaisen toteutusprojektin kannalta.

Prosessiuudistusten myötä päätettiin uudistaa myös tietojärjestelmäarkkitehtuuri, sillä liiketoimintaa piti pystyä tukemaan aina tietojärjestelmätasolta asti. Tämän uudistuksen alkumetreillä asiakas huomasi, että onnistuakseen uudistamaan arkkitehtuurinsa projektiin tarvittiin keskeiseen rooliin vahva integraattori, jonka vastuulla on toteuttaa ja vahtia kokonaisuutta.

8.1.1. Edeltävän järjestelmäarkkitehtuurin vaikutukset liiketoimintaan

Projektia edeltänyt järjestelmäarkkitehtuuri ei siis ollut enää nykyaikaisten vaatimusten mukainen, joten se ei pystynyt tukemaan tarvittavasti organisaation kasvua ja kehitystä. Käytetyt tietojärjestelmät ja niiden ominaisuudet olivat hyvin monenkirjavia, eikä kunnollista integraatioarkkitehtuuria ollut.



Kuva 8. Asiakkaan järjestelmäarkkitehtuuri ennen projektia

Edeltäneessä tietojärjestelmäkentässä keskeisessä roolissa olivat teollisuuden kunnossapidon järjestelmät (Kuva 8.). Nämä tietojärjestelmät ovat alalla pääsääntöisesti pitkäikäisiä, joten teknologisia ratkaisuja ei ole toteutettu uusimmilla teknologioilla. Järjestelmät ovat myös pitkälle tuotteistettuja, joten tuotteiden väliset sisäiset prosessit voivat vaihdella suuresti. Jokaisessa järjestelmässä on myös omat sisäiset tietokantansa, johon tietoa kerääntyy, eikä se ole lähtökohtaisesti yhteisessä käytössä.

Edellä mainituista seikoista johtuen yrityksen liiketoimintaprosessit pyörivät pääsääntöisesti yksittäisen kunnossapidon järjestelmän sisällä. Koska erilaisia kunnossapidon järjestelmiä oli yhtä aikaa käytössä useampia, järjestelmien väliset käsitteet ja toimintatavat olivat riippuvaisia siitä, mitä järjestelmää kulloisessakin tehtaassa käytettiin. Näitä järjestelmiä ei ollut myöskään integroitu keskenään, vaan ne kommunikoivat ainoastaan taloushallinnon ja laskutusprosessin kanssa. Osa

integraatioista oli toteutettu Point-to-Point -integraatioilla, joten näiden hallinta oli hyvin vaikeaa. Lisäksi tietoa siirrettiin kerran päivässä tapahtuvien eräajojen kautta, joten moderneja palveluita tai lähes reaaliaikaista tiedon hyödyntämistä oli vaikea toteuttaa.

Edellä kuvatulla järjestelmäarkkitehtuurilla oli suoria vaikutuksia työntekijöiden työhön ja toimintatapoihin. Vanhat järjestelmät eivät mahdollistaneet uusien mobiilisovellusten hyödyntämistä, lisäksi työntekijälle oleellinen tieto oli hajaantunut useaan eri järjestelmään. Tästä johtuen työntekijöillä ei ollut mahdollista saada reaaliaikaisesti työn tekemiseen tarvittavia tietoja. Nykyaikaisten mobiililaitteiden puuttumisen vuoksi työntekijöiden täytyi myös tehdä työvaiheiden ja työtuntien kirjaukset vasta työvuoron päätyttyä. Tästä seurasi, että tiimiesimiehien oli vaikea tehdä tarkkoja työsuunnitelmia, koska työtilanteiden seuranta tapahtui viiveellä. Organisaatiotason resursointi oli myös haasteellista, sillä tieto ei kulkenut tehtaiden välillä. Näin ollen työkuormaa ei voitu tasoittaa tulevien työntehtävien mukaisesti. Lisäksi varastonhallinta oli yksittäisen kunnossapidon järjestelmän vastuulla, joten tehtaat eivät tienneet toistensa varastosaldoja. Tämän johdosta jouduttiin säilyttämään suurempia varastomääriä ja tilaustuotteiden toimitusajat pitenivät.

8.1.2. Tietojärjestelmiin liittyneet ongelmat

Tarkasteltavan yrityksen lähtötilan tietojärjestelmiin liittyi myös joitain teknisiä haasteita. Kuten aikaisemmin todettiin, kyseiset kunnossapidon järjestelmät ovat hyvin pitkälle tuotteistettuja. Järjestelmätoimittajat toimivat osittain ainoastaan Suomen markkinoilla, minkä vuoksi järjestelmiä on voitu räätälöidä hyvinkin paljon asiakkaan tarpeisiin. Räätälöinneillä tarkoitetaan tässä yhteydessä sitä, että järjestelmien tietomalleja on voitu muuttaa tai laajentaa varsinaisesta toteutuksesta, jotta järjestelmiin on voitu tehdä kokonaan uusia toiminnallisuuksia. Nämä räätälöinnit heijastuivat välittömästi, kun projektissa alettiin suunnitella järjestelmien integroimista yhdeksi suureksi kokonaisuudeksi. Lisäksi tietojärjestelmien ylläpito oli asiakkaan puolelta hyvin

henkilöitynyttä, mikä lisäsi uudistusprojektiin kohdistuvia riskejä ja toisaalta vaikeutti tiedonsaantia projektin aikana.

8.2. Asiakkaan tavoitteet projektille

Asiakkaalla oli selkeät tavoitteet projektille, olihan kyseessä massiivinen tietojärjestelmäuudistus, jonka keskiöön ja tueksi hankittiin ensimmäistä kertaa integraatioarkkitehtuuri ja palveluväylä. Keskeisimpänä tavoitteena oli tietysti edellä kuvattujen tietojärjestelmien ja vanhan arkkitehtuurin puutteiden korjaus. Lisäksi liiketoiminnalla oli selkeät visiot siitä, mitä uudet järjestelmät ja toimintatavat voivat mahdollistaa.

Asiakkaan näkökulmasta koko projektin kantavana ajatuksena oli, että työntekoa pystyttäisiin tehostamaan tietojärjestelmien avulla. Työn tehostamista lähdettiin tavoittelemaan koko organisaation leveydeltä, aina tuotannollisista työntekijöistä, taloushallintoon ja johtoon asti. Tehtaassa työskentelevälle tämä näkyisi siten, että hän saisi käyttöönsä mobiililaitteen, johon integraatiokerros saisi toimitettua muista järjestelmistä kaikki työntekoon tarvittavat tiedot. Toiminta tehostuisi myös ongelmatilanteissa, sillä päätelaitteen ansiosta työntekijän olisi mahdollista hakea tietoa ilman, että hänen tarvitsisi siirtyä kiinteän päätteen luokse.

Esimiesten työnteko taas paranisi paremman työnohjauksen ja seurattavuuden myötä. Näiden lisäksi taloushallinnon työkuormaa pystyttäisiin vähentämään, sillä kaikki laskutus ja käytetyt resurssit kulkisivat sähköisesti integraatiokerroksen valvonnan alaisuudessa. Näin ollen tieto pysyisi yhtenäisenä ja laskuntarkistusta pystyttäisiin osittain automatisoimaan.

Liiketoiminnan tavoitteena taas oli vähentää kustannuksia, joita syntyy uuden asiakkuuden solmimisesta tietojärjestelmien näkökulmasta. Käytännössä tämä tarkoitti

integraatiokustannuksia, jotta uuden asiakkaan tietojärjestelmät saataisiin toimimaan yhteneväisesti jo olemassa olevien järjestelmien kanssa. Pienentyneet uusasiakashankinnan kustannukset taas toisivat säästöjä ja mahdollistaisivat organisaation nopeamman kasvun. Toisaalta koska käyttöön tulisi palveluihin perustava, globaali integraatioarkkitehtuuri, yritys kykenisi paremmin hankkimaan uusia asiakkaita uusilta maantieteellisiltä liiketoiminta-alueilta.

Projektilta tavoiteltiin myös materiaalinhallinnan ja ostoprosessin kokonaisvaltaista sähköistämistä ja uudistamista. Tällä pyrittiin siihen, että materiaalinhallinta yhtenäistettäisiin koko organisaatiossa. Nykyaikainen materiaalinhallinta yhdistettynä työnohjaukseen ja ennakkotöiden suunnitteluun tehostaisi varastonhallintaa ja vähentäisi varastokustannuksia. Varastokustannusten minimoinnin lisäksi asiakkaan tavoitteena oli ostoprosessin tehostaminen ja nykyaikaistaminen.

Ostoprosessin parantamisen lähtökohtana oli, että kun viestiliikenne ja tiedon hallinta tulee nykyaikaisen palveluväylän haltuun, voitaisiin prosessia sähköistää perustuen ajantasaisiin tavarantoimittajan hintoihin ja varastosaldoihin. Tällä saataisiin tehostettua työtä, koska työhön tarvittavien varaosien saldot ja toimitusaikatiedot olisivat aina saatavilla. Samalla vähentyisivät myös kustannukset, sillä ostoja tehtäisiin aina kilpailukykyisimmällä hinnalla. Aikaisemmin työhön liittyviä ostoja oli tehty työsuunnittelun ohessa, mikä oli hidastanut esimiesten työtä. Osa ostoista oli toteutettu vielä vanhanaikaisesti etsimällä tarvikkeita paikallisista varaosaliikkeistä tai puhelimen välityksellä. Tämä oli johtanut siihen, että kaikkia varaosia ei ollut saatavilla tarvittaessa tai niiden hankinta aloitettiin vasta siinä vaiheessa, kun varsinainen työnteke oli jo aloitettu.

Asiakkaalla oli myös selkeä visio siitä, millaisia hyötyjä järjestelmähanke tarjoaisi loppuasiakkaille. Integraatioarkkitehtuurin ollessa kaiken keskiössä, tarjoten nykyaikaiset rajapinnat tiedon jakelulle, mahdollistaa se lähes reaaliaikaisen tiedon jakelun myös asiakkaalle, esimerkiksi julkaisemalla käyttäjälle oleellisia tietoja ekstranet -palvelussa.

Ekstranetillä tarkoitetaan tässä yhteydessä yrityksen ja asiakkaan välistä suljettua verkkopalvelua. Laskutusprosessia pystytään myös selkeyttämään asiakkaille, koska kaikki työvaiheet ja kulutetut materiaalit ovat uudessa tietojärjestelmäkentässä sähköisessä muodossa.

Projektin yksi suurimmista tavoitteista oli raportoinnin parantaminen, ja joiltain osin myös mahdollistaminen. Vanhat järjestelmät olivat keskustelleet niin huonosti keskenään, että kunnollista raportointia oli ollut hyvin vaikea toteuttaa. Asiakkaan toiveena olikin, että palveluväylän yhteyteen saataisiin toimiva BAM-ratkaisu, jolloin pystyttäisiin ketterämmin reagoimaan tarvittaviin muutoksiin. Lisäksi seuraavana kehitysaskeleena olisi kunnollisen tietovaraston rakentaminen, jotta sähköisessä muodossa olevaa tietoa voitaisiin raportoida nykyistä kattavammin.

8.3. Haasteet integraatioarkkitehtuuria muutettaessa

Integraatioprojekteihin ja integraatiotyöhön liittyy aina omat haasteensa, kun joudutaan työskentelemään vahvasti yksittäisten järjestelmien ja liiketoiminnan yhteen sovittelijana. Tämän tapaustutkimuksen projektissa nämä haasteet ja riskit korostuivat entisestään. Koska toimittiin ympäristössä, jossa valtaosa aikaisemmasta tietojärjestelmäkentästä päivitettiin tai korvattiin kokonaan uusilla järjestelmillä enemmän tai vähemmän Big Bang -tyyppisesti.

Tämän seurauksena muuttuvien järjestelmien määrittelyt olivat hyvin erilaisissa vaiheissa, riippuen siitä kuinka suurta muutosta uudistus vaati kyseiseltä järjestelmältä. Kaikella tällä oli suorat vaikutukset myös integraatioarkkitehtuuriin ja varsinaiseen integraatiotyöhön, sillä rajapintoja ja palveluita jouduttiin määrittelemään ja osittain toteuttamaan vielä muuttuvien järjestelmien tarpeisiin. Integraatiotyössä ei saanut liikaa turvautua vanhoihin vesiputousmallin mukaisiin menetelmiin, vaan täytyi pystyä toimimaan mahdollisimman ketterästi.

Kyseinen projekti oli myös siinä mielessä erikoinen, että asiakas uudisti yhdessä tietojärjestelmien kanssa lähes kaikki liiketoimintaprosessinsa. Tällä oli suoria vaikutuksia myös integraatiotyöhön sillä, kun asiakkaan työntekijöiden päivätyö muuttuu merkittävästi, on liiketoiminnankaan määrittelyvaiheessa hyvin vaikea nähdä todellisia työtehtäviä ja näistä muodostuvia tarpeita. Usein tällainen tilanne johtaa siihen, että integraatiokerroksen palveluita muodostetaan tietojärjestelmien tarpeisiin, eikä todellisiin liiketoimintatarpeisiin, sillä järjestelmien määrittelemät rajat on yleensä helpompi havaita. Tätä välttääkseen integraatioarkkitehtuurille muodostui tavanomaista suurempi vastuu koko projektin valmomisesta ja hallinnasta.

8.4. Toteutunut arkkitehtuuri

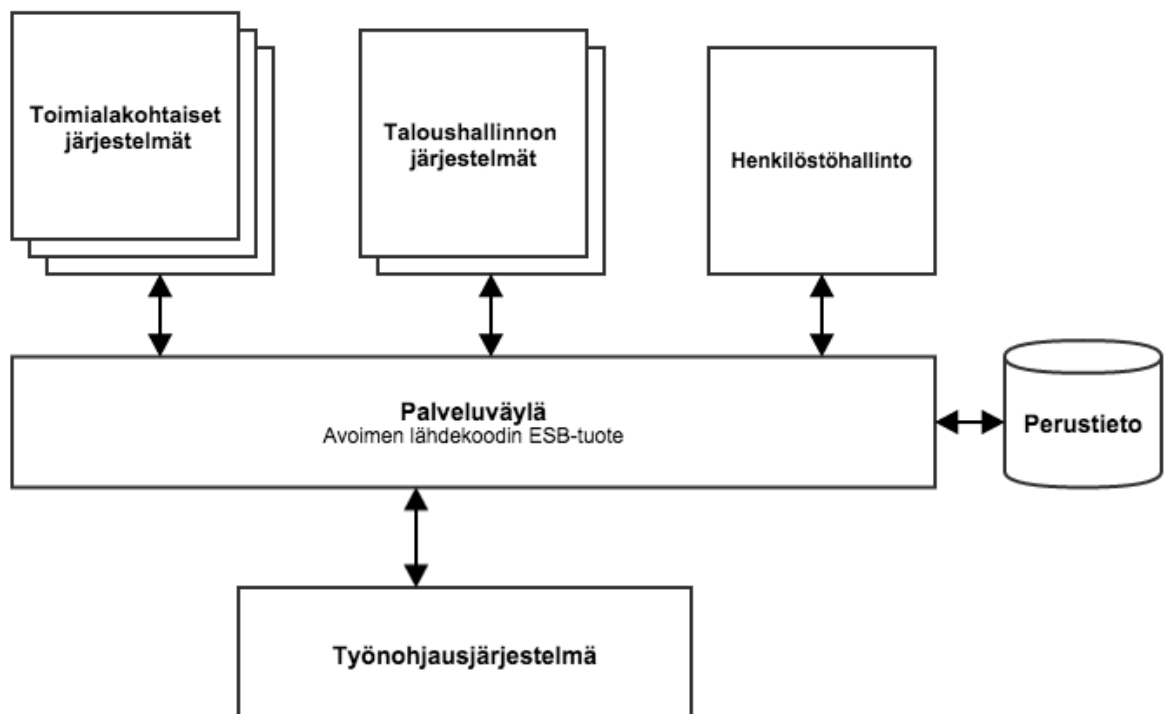
Tapaustutkimuksen projektissa päädyttiin toteuttamaan integraatioarkkitehtuuri (Kuva 9.), joka mukaili hyvin pitkälle palvelukeskeisen arkkitehtuuriin pääpiirteitä. Valintaan päädyttiin koska nähtiin, että yleiskäyttöisten palveluiden avulla pystyttäisiin parhaiten palvelemaan asiakasta ja sen monimuotoisia liiketoimintatarpeita.

Täten jokainen asiakkaan keskeisistä liiketoiminnan kokonaisuuksista pyrittiin mallintamaan yleispäteviksi palveluiksi. Palvelut on taas sidottu liiketoimintakokonaisuuksiin aikaisemmin esitellyn kanonisen mallin avulla. Kanonisen mallin tehtävänä on pitää liiketoimintakriittiset tiedot yleisessä formaatissa, jolloin tietomallin ja tiedon ei tarvitse olla riippuvainen yhdestäkään lähde- tai kohdetietojärjestelmästä.

Varsinaiset palvelut toteutettiin avoimen lähdekoodin palveluväylä- ja integraatiotuotteen päälle. Kyseiseen tuotteeseen päädyttiin sen mukautuvuuden ja monipuolisten rajapintojen vuoksi. Palvelukeskeisen arkkitehtuurin mukaisesti palvelut itse ovat vastuussa integraatiokerroksen logiikasta ja tiedon välityksestä tarvittaville osapuolille.

Järjestelmäkohtaiset tekniset yksityiskohdat, kuten tietoliikenneprotokollat, salasanat, sekä tietoformaatit on piilotettu varsinaiseen palveluiden toteutukseen. Näin ollen varsinaiset kohde- ja lähdejärjestelmät pysyvät palvelukeskeisen arkkitehtuurin periaatteiden mukaisesti tietämättöminä toisistaan. Järjestelmien yhdistäminen jää palveluväylään toteutettavan rajapintakerroksen vastuulle.

Palveluväylän rinnalle koettiin myös tarpeelliseksi rakentaa järjestelmä, joka tallettaisi kaiken keskeisen organisaation ja sen liiketoimintaan liittyvän tiedon. Siksi palveluväylän yhteyteen tehtiin kiinteä ratkaisu perustiedon tallennukseen.



Kuva 9. Tapaustutkimuksen projektissa toteutunut integraatioarkkitehtuuri

8.4.1. Perustieto

Perustieto -käsitteellä (Master Data) tarkoitetaan usein tietoa, joka on elintärkeää organisaation toiminnalle. Perinteisesti tällaisena on nähty esimerkiksi asiakas-, toimittaja-

, organisaatio- ja tuotetietoja sekä sitä, miten nämä liittyvät toisiinsa. Jokainen näistä perustiedon yläkäsitteistä sisältää sellaista tietoa, joka on keskeistä koko organisaation liiketoiminnalle, mutta myös hyvin kriittistä muille operationaalisille järjestelmille ja liiketoimintayksiköille. Perustieto tarkoittaa siis tietoa kohteista, jotka on hyväksytty ja jaettu koko liiketoiminnan yli. [Berson and Dubov, 2011].

Luonteeltaan perustieto on staattista, sillä liiketapahtumatieto tieto ei ole perustietoa. Liiketapahtumatieto tiedolla tarkoitetaan tässä yhteydessä esimerkiksi organisaation myyntitietoja. Tästä syystä perustiedon keskeiset käsitteet ovat usein samanlaisia organisaation koosta tai liiketoiminta-alasta huolimatta. Käsitteiden sisältämät tiedot ja tietomäärät voivat vaihdella hyvinkin paljon, sen mukaan millä tarkkuustasolla liiketoiminta kuvaa esimerkiksi yrityksen tuotteiden tietoja. Voidaankin sanoa, että perustiedon tietomalli kuvaa lähes sellaisenaan yrityksen toiminnan [Dreibelbis *et al.*, 2008].

8.4.1.1 Perustieto osana integraatioarkkitehtuuria

Mitä tekemistä perustiedolla on palvelukeskeisen arkkitehtuurin ja palveluväylän kanssa? Usein perustietoa ruvetaan kehittämään raportoinnin näkökulmasta, mutta tässä projektissa otettiin lähtökohdaksi, että laadukkaasta perustiedosta hyötyvät kaikki organisaation järjestelmät ja prosessit. Peruseriaatteiltaan perustiedon yleiskäyttöinen tietomalli vastaa myös palvelukeskeisen arkkitehtuurin kanonisen tietomallin ajattelutapaa. Perustiedossa keskiöön nostetaan organisaation, ei yksittäisten järjestelmien käsite- ja tietomalli.

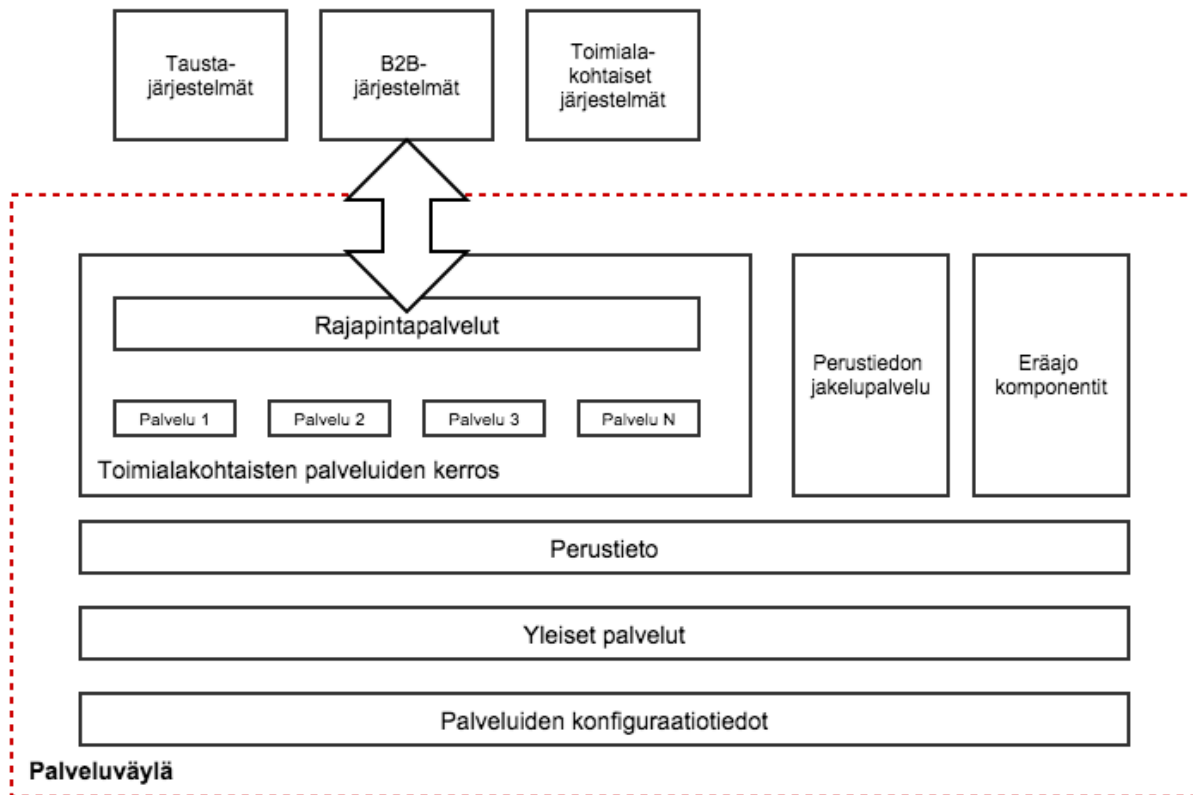
Näin ollen perustiedon pitää olla kiinteässä vuorovaikutuksessa palveluväylän kanssa, jotta tiedon saavutettavuus olisi mahdollisimman hyvää. Tässä mallissa yksittäisen perustietokäsitteen pääjärjestelmän vastuulla on tietojen ajantasaisuuden varmistus ja syöttö perustietojärjestelmään, kun taas palveluväylän vastuulle jää tiedon jakelu.

Edellä kuvatulla suunnittelumallilla saavutetaan selkeitä etuja integraatiokehityksessä. Integraatiosanomien sisällössä ei välttämättä tarvitse olla kuin keskeiset tunnistetietueet, sillä tarvittavat järjestelmäkohtaiset kohdistukset ja tiedon rikastaminen voidaan toteuttaa perustiedon pohjalta. Toisaalta perustietojärjestelmällä saadaan myös vähennettyä yksittäisen perustietokäsitteen pääjärjestelmän kriittisyyttä ja täten vähennettyä viestiliikennettä yksittäisten järjestelmien välillä.

Esimerkkinä voidaan pitää yrityksen henkilötietojärjestelmää, joka toimii organisaation henkilötiedon pääjärjestelmänä. Integraatiokerroksen vastuulla on hakea perustietojärjestelmään organisaation ajantasaiset henkilötiedot, ja jakaa näitä tietoja niitä tarvitseville järjestelmille. Perinteisessä mallissa integraatiokerros olisi hakenut henkilötietojärjestelmän tiedot jokaisen haun yhteydessä, ja samalla mahdollisesti kuormittanut järjestelmää turhaan.

8.4.2. Palveluväylä integraatioarkkitehtuurin keskiössä

Kuten aiemmin on todettu, palveluväylä on yksi SOA-arkkitehtuurin keskeisimmistä komponenteista. Sen päätehtävänä on toteuttaa arkkitehtuurin viestinvälitys. Tapaustutkimuksen projektissa palveluväylän arkkitehtuuri (Kuva 10.) koostui erilaisista palveluista, jotka taas voidaan jakaa useisiin eri kerroksiin käyttötapansa ja kohteensa perusteella.



Kuva 10. Tapaustutkimuksen projektissa toteutunut palveluväyläarkkitehtuuri

Arkkitehtuurisesta näkökulmasta tavoitteena oli tehdä palveluväylästä mahdollisimman ketterä, mikä mahdollistaisi myös toiminnallisuuksien uudelleenkäytön. Tämän vuoksi toimialakohtaiset palvelut tuotettiin omaksi palvelukerrokseksi, joka sisältää yleismaailmalliset palvelut kaikista tunnistetuista liiketoimintakäsitteistä. Nämä palvelut on nivottu yhteen ja käytettäväksi erillisten rajapintapalveluiden kautta.

Rajapintapalveluiden vastuulla on huolehtia integraatiosta tiedonvälityksen tasolla ja samalla suorittaa mahdollinen järjestelmäkohtainen tiedon ja teknologian muunnos. Teknologian muunnoksella tarkoitetaan sitä, että loogiseen palveluun pystytään kytkeytymään vaihtelevilla rajapintateknologioilla. Toimialakohtaisten palveluiden lisäksi hyvin keskeisessä roolissa on edellisessä kappaleessa kuvattu perustietojärjestelmä, jossa ylläpidetään kaikkia liiketoiminnan keskeisiä käsitteitä. Kiinteän siten vuoksi perustieto

voidaankin nähdä loogisesti kuuluvaksi osaksi palveluväylää, vaikkakin kyseessä on erilliset teknologiat ja palvelimet.

Edellä mainittujen lisäksi palveluväylälle toteutettiin erilliset eräajokomponentit ja perustiedon jakelupalvelut. Näiden avulla palvelukerros pystyy hakemaan ja vastaanottamaan pääjärjestelmien tietoa, ja samalla tallettamaan sen perustietojärjestelmään. Perustietojärjestelmän tietoja voidaan myös jaella eteenpäin tehtyjen palveluiden avulla. Erilliset palvelut mahdollistavat myös tarkemman kontrollin ja valvonnan eräajoin tapahtuvaan tiedonsiirtoon

Palveluväylälle toteutettiin myös erillinen yleispalvelukerros, joka koostuu yhteisesti käytetyistä toteutuskomponenteista. Tämän ansiosta palveluväylän on mahdollista uudelleenkäyttää ja jakaa muun muassa samoja apupalveluita, kirjastoja, sekä tiedonmuuntimia. Kaikkien palveluväylän toiminnallisuuksien tueksi toteutettiin vielä erillinen kerros konfiguraatietiedolle, joka sisältää tarvittavat tiedot palveluväylän toimimiselle. Tämän erillisen kerroksen avulla pystytään helposti ottamaan käyttöön lähde- ja kohdejärjestelmiä ilman, että varsinaiseen integraatiototeutukseen tarvitsee puuttua.

8.5. Mitä integraatioarkkitehtuurilla saatiin aikaan?

Tapaustutkimuksen kohteena olevan tietojärjestelmäprojektin ensimmäisestä tuotantoversiosta on aikaa noin vuosi, ja jo tänä aikana on havaittu selkeitä muutoksia, joita on saavutettu sekä teknisellä, että liiketoiminnan tasolla. Palveluväylä on koko järjestelmäarkkitehtuurin ydin ja suuremman kokonaisuuden mahdollistaja, mutta siitä saatavat nettohyödyt realisoituvat loppukäyttäjälle vasta muiden järjestelmien kautta. Tämä asetelma on hyvä pitää mielessä integraatioratkaisun tuloksia arvioitaessa.

Teknologisesta näkökulmasta yrityksen toimintatavat ja tulevaisuuden mahdollisuudet muuttuivat huomattavasti nykyaikaisemmiksi. Teknisesti osa uusista asiakkuuksista ja järjestelmistä on ollut palveluväylän myötä ylipäättään mahdollista integroida osaksi organisaation toimintaa. Ratkaisut, jotka olivat käytössä ennen projektia eivät tukeneet kaikkia loppuasiakkaiden tai taustajärjestelmien tarvitsemia rajapintoja. DeLonen ja McLeanin [2003] tietojärjestelmien onnistumismallin mukaisesti tämä teknologinen kehittyminen voidaan nähdä järjestelmien laadun paranemisena. Esimerkkinä tällaisesta on www-pohjaiset rajapinnat, joita järjestelmät voivat käyttää tietojen syöttämiseen ja hakemiseen.

Palveluväylä myös sitoo kaikki organisaation käyttämät tietojärjestelmät yhden valvovan järjestelmän alaisuuteen. Organisaation IT-osastolla tämä näkyy siten, että monitorointi ja järjestelmien välisten integraatioiden hallinta on aiempaa helpompaa. Palveluväylän yhteyteen toteutetun valvontatyökalun kautta IT-osasto pystyy näkemään, jos esimerkiksi jokin teollisuuden kunnossapidon järjestelmä ei ole pystynyt tuottamaan sovitussa aikataulussa integraatioihin tarvittavia tietoja. Tämän ansiosta vian aiheuttaneen järjestelmän selvittäminen on huomattavasti aikaisempaa helpompaa ja nopeampaa. Tiedon laadulla ja saavutettavuudella on siis saatu parannettua IT-osaston käyttäjätyytyväisyyttä.

Integraatioarkkitehtuurin edut ovat myös realisoituneet liiketoiminnan tasolle. Kun uusia asiakkuuksia solmitaan, käydään läpi erilaiset tietojärjestelmä- ja integraatiovaihtoehdot. Aikaisemmin nämä keskustelut ovat olleet hyvin aikaa vievä työvaihe, sillä selkeää prosessia ja kunnollista integraatioarkkitehtuuria ei ole ollut ohjaamassa keskusteluja. Arkkitehtuurin ja palveluväylän ansiosta integraatiokeskustelu on nopeutunut ja helpottunut, koska organisaatio voi jo olemassa olevien palveluiden myötä kertoa integraatioiden peruspiirteistä ja rajoitteista. Nämä selkeät ääriverit vähentävät myös sitä riskiä, että neuvotteluissa sovittaisiin sellaisista palveluista, joita palveluväylä ei

mahdollistaisi ilman suuria muutoksia jo olemassa olevaan toteutukseen. Toisaalta myös joustaminen näissä rajoituksissa on helpompaa.

Edellä mainitut seikat ovat vähentäneet uuteen asiakkaaseen liittyviä aloituskustannuksia, sillä asiakassuhteen käynnistyminen on ollut nopeampaa ja integraatioihin liittyviä työvaiheita on ollut aikaisempaa vähemmän. Helpoimmat tapaukset ovat sellaisia, joissa palveluväylään integroituu jokin tietojärjestelmätuote useampaan kertaan. Tällaisissa tapauksissa integraatiotyöksi muodostuu ainoastaan palveluväylän konfigurointitietojen päivitys. Loppuasiakkaalle integraatioprosessin nopeutuminen näkyy palvelun laadun paranemisena, joka taas puolestaan vähentää asiakkaalle kohdistuvia integraatiokustannuksia.

Palveluväylälle toteutettu toimialakohtaisten palveluiden kerros on tuonut yritykselle mahdollisuuden uudenlaiseen yhteistyöhön niiden toimittajien kanssa, jotka vastaavat teollisuuden kunnossapidon järjestelmistä. Vakioituneiden palveluiden ja rajapintojen ansiosta voidaan sopia kiinteästi hinnoiteltuja, räätälöityjä paketteja, jotta loppuasiakkaan järjestelmä saadaan integroitua toteutettuun integraatiokerrokseen. Tämä on omalta osaltaan lisännyt palveluväylään liittymisen houkuttelevuutta ja samalla vähentänyt integraatiokustannuksia. Kaikki tämä näkyy järjestelmien ja palveluiden laadun paranemisena, jota myös DeLonen ja McLeanin [2003] onnistumismalli painottaa.

Edelle mainittujen lisäksi palveluväylä on mahdollistanut organisaatiolle tavan seurata tehokkaasti työvaiheiden etenemistä. Tätä mahdollisuutta on jalostettu pidemmälle siten, että organisaatio on alkanut käyttämään avoimet kirjat -hinnoittelumallia (open-book accounting). Avoimet kirjat -mallilla tarkoitetaan sellaista prosessia, jossa yhteistyötä tekevät yritykset avaavat omat kustannusrakenteensa joko osittain tai kokonaan [Tenhunen, 2006].

Loppuasiakkaille muutos on näkynyt parempana palveluna ja läpinäkyvämpänä laskutusprosessina. Aikaisemmin loppuasiakkaita oli laskutettu erilaisten työmääräarvioiden pohjalta, mutta palveluväylän ja uuden mobiilisovelluksen ansiosta työntekijät ovat voineet kuitata työvaiheensa tehdyksi työnteon lomassa. Tämän tiedon ansiosta laskulle kirjautuu oikeasti toteutuneet työvaiheet, joten asiakkaan ei tarvitse maksaa turhasta. Voidaankin siis sanoa, että tiedon laatu on myös parantunut palvelun laadun ohella. Avoin hinnoittelumalli ajaa myös loppuasiakasta tukemaan ja parantamaan työntekijöiden työympäristöä, sillä tehostuneet työvaiheet heijastuvat välittömästi loppulaskulle.

Reaaliaikaisen viestiliikenteen ja perustiedon tarjoaman ajantasaisen liiketoimintatiedon ansiosta on voitu myös tehokkaammin raportoida organisaation toimintaa. Vaikka varsinaista tietovarastoa ei ehditty toteuttaa projektin aikataulun puitteissa niin silti tiedon laatu on parantunut huomattavasti varsinaisissa operatiivisissa järjestelmissä. Järjestelmäkentän totaalisen muutoksen myötä organisaatio on voinut myös raportoida sellaisia asioita, joista aikaisemmin ei jäänyt minkäänlaista sähköistä jälkeä. Tämä syy-seuraussuhde saa tukensa myös tietojärjestelmien onnistumismallista, sillä tiedon ja järjestelmien laadun paraneminen lisää järjestelmien käyttöä ja siitä saatavia nettohyötyjä. Esimerkkinä tällaisesta uudesta raportointitarpeesta on yksittäiseen huoltotapahtuman vaiheeseen käytetty aika.

Oleellisin asia projektin jälkeen on ollut kuitenkin se, että tietojärjestelmäremontin avulla organisaatio onnistui yhtenäistämään oman palvelutuotantonsa. Tämän toteutumisessa palveluväylällä on ollut kriittinen rooli. Organisaation on siksi nykyään huomattavasti helpompi antaa realistinen palvelulupaus loppuasiakkaalle. Pääsääntöisesti tämän on mahdollistanut se, että tietojärjestelmien ja integraatioiden laatu ja toimintatapa on pystytty lähes täysin vakiomaan.

Edellä kuvatut johtopäätökset voidaan siis kuvata myös DeLonen ja McLeanin [2003] tietojärjestelmien onnistumismallin avulla. Palveluväylän ja perustietojärjestelmän avulla muiden tietojärjestelmien käyttämä liiketoimintatieto on yhtenäistynyt, parantunut ja muuttunut reaaliaikaisemmaksi. Tämä on puolestaan parantanut raportoinnin kautta saatavia tuloksia. Käytettävissä olevan laadukkaan tiedon lisääntyminen on helpottanut myös IT-ylläpidon arkea. Nykyaikainen palveluväylä taas toi muille tietojärjestelmille mahdollisuuden alkaa hyödyntää viimeisimpiä rajapintateknologioita, millä saatiin parannettua järjestelmien yleistä laatua. Selkein muutos on kuitenkin tapahtunut palveluiden laadussa. Tämä voidaan havaita sekä yksittäisen tietojärjestelmän tasolla, että kerrannaisvaikutusten kautta organisaatiossa. Teollisuuden palveluita tuottavan yrityksen on nyt mahdollista tuottaa sellaisia palveluita, jotka ennen tarkasteltavaa projektia olivat mahdottomia toteuttaa. Loppuasiakkaalle asiat näkyvät parempana palveluna ja yritys houkuttelevampana kumppanina.

Integraatiot ja integraatioprosessit ovat luonteeltaan muita operatiivisia toimintoja tukevia toimintoja, niin tässäkin tapaustutkimuksessa. Näin ollen palveluväylän avulla on saatu nostettua muiden tietojärjestelmien käyttöastetta ja samalla hyödynnettyä tietojärjestelmien potentiaali aiempaa paremmin. Kuvaavana esimerkkinä voidaankin pitää sitä, että nykyisin tehtaan työntekijä saa tarvitsemansa tiedot älypuhelimeen, kun aikaisemmin asioita piti kirjata käsin ylös ja tarkastaa yhteisiltä päätteiltä. Väylän ansiosta syntynyttä tietoa käytetään myös liiketoimintaa ja asiakassuhteita ohjaavana tekijänä. Tämä kaikki näkyy DeLonen ja McLeanin [2003] mallin mukaisesti organisaation saavuttamina nettohyötyinä. Kokonaisuudessaan parannukset lisäävät käyttäjätyytyväisyyttä ja tietojärjestelmien käyttöä, joten palvelukerrokseen ja muihin tietojärjestelmiin investointi on jatkossakin kannattavaa.

8.6. Ennakoimattomat vaikutukset ja mahdolliset parannusehdotukset

Kuten työssäni olen koettanut kuvata, tapaustutkimuksen kohteena oleva organisaatio uudisti tietojärjestelmiään hyvin voimakkaasti lyhyessä ajassa. Tämä aiheutti joihinkin kehityskohteisiin aikataulupaineita ja viivästymisiä. Joitakin projektin osa-alueita jouduttiin lykkäämään myöhempiä vaiheita varten, sillä asiakkaalla ei ollut tarvittavia resursseja toteuttaa useaa kehitystyötä yhtäaikaisesti.

Kaikkien positiivisten muutosten lisäksi projektista tuli joitakin sivuoireita, joita ei osattu ennakoida projektia suunniteltaessa. Yksi tällaisista on ollut vanhan IT-osaston henkilömäärän riittämättömyys. Aikaisemmin tietojärjestelmäkenttä koostui muutamasta teollisuuden kunnossapidon järjestelmästä ja taloushallinnon järjestelmästä, joita voitiin hallita pienemmillä henkilöresursseilla kuin uudistunutta tietojärjestelmäkenttää. Projektin myötä kriittisten operatiivisten järjestelmien määrä kaksinkertaistui, joten IT-osasto on joutunut toimimaan resursseihinsa nähden ylikuormalla.

Toinen lisätyötä aiheuttanut tekijä on ollut tiedon laatu. Järjestelmämuutoksen keskeisenä ajatuksena oli se, että tehtaiden työntekijät itse kirjaavat tuntinsa ylös mobiililaitteen avulla. Varoituksista huolimatta organisaatio ei ole ottanut tiedon laatua tarpeeksi kriittiseksi tekijäksi, minkä seurauksena esimerkiksi taloushallinto on joutunut käyttämään enemmän aikaa laskujen tarkastukseen ja hyväksyntään. DeLonen ja McLeanin [2003] tietojärjestelmien onnistumismallin mukaisesti virheellinen tieto heijastuu suoraan loppukäyttäjän kokemaksi negatiiviseksi kokemukseksi ja saavutettavat hyödyt vähentyvät. Osittain tiedon heikko laatu on johtunut siitä, että työntekijöiden työtä tukevat järjestelmät ja tuntikirjausperiaatteet ovat muuttuneet täysin aikaisempiin käytäntöihin verrattuna. Siksi käyttäjien perehdyttäminen ja vallitsevien käytäntöjen juurruttaminen organisaation vie vielä aikaa.

Aikataulun muuttumisen ja resurssien puutteiden johdosta joitain projektin osakokonaisuuksia jouduttiin lykkäämään tulevaisuuteen. Merkittävimpinä näistä ovat materiaalinhallinta, sähköiset ostot, tietovarasto ja BAM-ratkaisu. Osakokonaisuudet ovat itsessään jo suuria tietojärjestelmähankkeita, joten niiden toteuttaminen vaatii suuria investointeja, mutta niiden avulla voidaan tehostaa organisaation toimintaa.

Erityisesti materiaalihallinnan ja sähköisten ostojen toteutuminen toisi asiakkaalle selvää kilpailuetua ja kustannussäästöjä, koska ostot pystyttäisiin suorittamaan aina kilpailukykyisimmällä markkinahinnalla, mutta samalla varastointikustannuksia saataisiin pienennettyä. Tietovaraston ja BAM:n avulla organisaatio pystyisi taas kehittämään raportointiaan vielä nykyistäkin paremmaksi. Raportoinnin kehittyttyä organisaatio voisi löytää kehityskohteita, joita parantamalla voisi saavuttaa merkittäviä kustannussäästöjä ja uusia palveluita, joita asiakkaat tarvitsevat.

9. Yhteenveto

Nykyään tietoa ja sitä käsitteleviä tietojärjestelmiä on enemmän kuin koskaan aikaisemmin, eikä vaikuta siltä, että kehityksen nykyinen suunta olisi muuttumassa. Samaan aikaan nykyaikaisessa yritystoiminnassa eletään yhä voimakkaammin verkostoitumisen aikakautta, joten yritysten liiketoimintaprosessit voivat koostua yhteistyöstä kumppanien ja alihankinnan kanssa. Täten yksittäiseen tietojärjestelmään kertynyt tieto ei ole sellaisenaan kovin arvokasta, ellei sitä pystytä jakamaan ja hyödyntämään muiden järjestelmien ja toimijoiden kesken. Tarve tiedon yhdistämiselle, integroimiselle, on aiempaa vahvempi.

Integraation luonne on vuosien myötä muuttanut muotoaan. Aikaisemmin kyse oli enemmänkin teknisestä integraatiosta, esimerkiksi viestin välityksestä kahden järjestelmän välillä. Nykyään integraatio tarkoittaa paljon muutakin, kuten liiketoimintaprosessien hallintaa. Muutos on asettanut omat vaatimuksensa erilaisille integraatioarkkitehtuureille ja malleille, jotta ne voisivat vastata nykypäivän haasteisiin.

Palvelukeskeinen arkkitehtuuri on suunnittelutapana pystynyt vastaamaan tähän haasteen. Järjestelmäriippumattomat palvelukuvaukset, uudelleen käytettävät palvelut, löyhät sidokset ja yhteentoimivuuden periaate mahdollistavat palveluiden kehittämisen, jakelun ja aiempaa monipuolisemman käytön. Nykyaikainen arkkitehtuuri perustuukin palveluiden jakamiseen eri järjestelmien välillä. Palveluiden kattavaan jakeluun tarvitaan kuitenkin jokin keskeinen järjestelmä, esimerkiksi palveluväylä. Nykyaikaista integraatioarkkitehtuuria olisikin hyvin vaikea toteuttaa ilman palveluväylän kaltaista joustavaa ja komponentteihin perustuvaa kaiken keskiössä olevaa järjestelmää.

Osana tätä tutkielmaa havainnoitiin tapaustutkimuksena integraatioprojektia, jossa pyrittiin täyttämään teollisuuden palveluyrityksen integraatioarkkitehtuurin tarpeita ja

vaatimuksia. Projektin aikana toteutettiin palvelukeskeistä arkkitehtuuria mukaileva integraatioarkkitehtuuri, jonka ytimenä on avoimen lähdekoodin palveluväylätuote ja sen yhteyteen rakennettu perustietojärjestelmä. Tämän perustietojärjestelmän avulla pyrittiin huolehtimaan liiketoimintatiedon ajantasaisuudesta. Edellä kuvatulla ratkaisulla pystyttiin vastaamaan asiakasyrityksen vaatimuksiin palvelutuotannon yhtenäistämisestä ja vähentämään uusasiakkaiden kohdalla käytävää integraatiokeskustelua ja varsinaista integraatiotyötä. Loppuasiakas taas näkee muutoksen parempana palveluna ja tuoreimpina teknologisina ratkaisuin, jotka eivät aseta rajoitteita varsinaiselle liiketoiminnalle.

Tutkimuksen aikataulu ei mahdollistanut laajempaa, kvantitatiivista tutkimusta, sillä kohdeyrityksen luonteen takia aineistoa olisi ollut saatavilla vasta usean vuoden kuluttua. Jatkotutkimuksessa olisikin hedelmällistä selvittää faktatietoon perustuen kuinka paljon tehokkaammin uusi integraatioarkkitehtuuri mahdollistaa uusien asiakkaiden liittymisen palveluväylän yhteyteen. Lisäksi olisi hyvä selvittää millaisiin kustannussäästöihin sähköisillä kaupankäyntiputkilla ja nykyaikaisella materiaalinhallinnalla voitaisiin päästä teollisuuden palveluita tarjoavan yrityksen tapauksessa.

Viiteluettelo

- [Bennet, 1995] Keith Bennet, Legacy systems: Coping with success. *IEEE Software*, **12** (1995), 19–23.
- [Berson and Dubov, 2011] Alex Berson and Larry Dubov, *Master Data Management and Data Governance*. The McGraw-Hill Companies, 2011
- [Bussler, 2010] Christoph Bussler, *B2B Integration*. Springer, 2010
- [Cassel and Symon, 2004] Catherine Cassell and Gillian Symon, *Essential Guide to Qualitative Methods in Organisational Research*. Sage, 2004
- [DeLone and McLean, 1992] William DeLone and Ephraim McLean, Information systems success: The quest for the dependent variable. *Information Systems Research*. **3** (1992), 60-95.
- [DeLone and McLean, 2003] William DeLone and Ephraim McLean, The DeLone and McLean model of information systems success: A ten-year update. *Journal of Management Information Systems* **19** (2003), 9 - 30.
- [Dreibelbis et al., 2008] Allen Dreibelbis, Eberhard Hechler, Ivan Milman, Martin Oberhofer, Paul van Run and Dan Wolfson, *Enterprise Master Data Management - An SOA Approach to Managing Core Information*. IBM, 2008
- [Engel et al., 2013] Robert Engel, Worarat Krathu, Christian Pichler, Marco Zapletal and Hannes Werthner, Towards EDI-Based Business Activity Monitoring. Teoksessa Enterprise Distributed Object Computing Conference Workshops, *IEEE* **17** (2013), 158-162.
- [Erl, 2008] Thomas Erl, *Service-Oriented Architecture – Concepts, Technology, and Design*. Prentice Hall, 2008
- [Hevner et al., 2004] A.R. Hevner, S.T. March, J. Park and S. Ram, Design science in information systems research. *MIS Quarterly* **28** (2004), 75-105.

- [Hohpe, 2012] Gregor Hohpe, *Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions*. Pearson Education, 2012
- [Josuttis, 2007] Nicolai M. Josuttis, *SOA in Practice, the Art of Distributed System Design*. O'Reilly Media, 2007
- [Järvinen ja Järvinen, 1996] Pertti Järvinen ja Annikki Järvinen, *Tutkimustyön metodeista*. Opinpaja, Tampere, 1996.
- [Kang and Han, 2008] Jin Gu Kang and Kwan Hee Han, A business activity monitoring system supporting real-time business performance management. In: *International Conference on Convergence and Hybrid Information Technology*, **3** (2008), 473-487.
- [Mykkänen *et al.*, 2003] Juha Mykkänen, Jari Porasmaa, Juha Rannanheimo ja Mikko Korpela, A process for specifying integration for multi-tier applications in healthcare. *International Journal of Medical Informatics* **70** (2003), 172-182.
- [Saboniha *et al.*, 2012] Nazanin Sabooniha, Danny Toohey and Kevin Lee, An evaluation of hospital information systems integration approaches. In: *Proc. of ICACCI* (2012), 498-504.
- [Schmidt and Lyle, 2010] John G. Schmidt and David Lyle, *Lean Integration – An Integration Factory Approach to Business Agility*. Addison-Wesley, 2010
- [Schön, 1983] Donald A. Schön, *The reflective practitioner, how professionals think in action*. Basic Book Inc., 1983
- [Shanin and Samea, 2010] Arash Shanin and Monireh Samea, Developing the models of service quality gaps: a critical discussion. *Business Management and Strategy* **1** (2010)
- [Siltanen, 2004] Juha Siltanen, Tietoarkkitehtuurin perustuva sovellusintegraatiometodi – tapaus Tieliikelaitos. Pro gradu -tutkielma, Tietojenkäsittelytieteiden laitos, Tampereen yliopisto, 2004.
- [Tenhunen, 2006] Jarkko Tenhunen, Johdon laskentatoimi kärkiyritysverkostoissa. Väitöskirja, Lappeenrannan teknillinen yliopisto, 2006
- [Trowbridge *et al.*, 2004] David Trowbridge, Ulrich Roxburgh, Gregor Hoppe, Dragos Manolescu and E.G Nadhan, *Integration Patterns*. Microsoft, 2004

- [UN/EDIFACT, 2014] United Nations, Directories for Electronic Data Interchange for Administration, Commerce and Transport. Saatavilla: <http://www.unece.org/fileadmin/DAM/trade/untdid/d14a/trmd/trmdi2.htm> (käytetty: 23.9.2014)
- [Tähtinen, 2005] Sami Tähtinen, *Järjestelmäintegraatio - Tarve, vaihtoehdot, toteutus*. Talentum, Helsinki, 2005
- [Wisniewski, 2001] Mik Wisniewski, Using SERVQUAL to assess customer satisfaction with public sector services. *Managing Service Quality* **11** (2001), 380-388
- [Yin, 1989] Robert K. Yin, *Case Study Research: Design and Methods*. SAGE Publications, Beverly Hills Ca., 1989